NUNC COGNOSCO EX PARTE

TRENT UNIVERSITY
LIBRARY

ELECTRONIC ANALOG AND HYBRID COMPUTERS

ELECTRONIC ANALOG AND HYBRID COMPUTERS

Second, Completely Revised Edition

GRANINO A. KORN, Ph.D.
Professor of Electrical Engineering, University of Arizona

THERESA M. KORN, M.S.

McGRAW-HILL BOOK COMPANY
New York St. Louis San Francisco Düsseldorf Johannesburg
Kuala Lumpur London Mexico Montreal New Delhi
Panama Rio de Janeiro Singapore Sydney Toronto

TK7888 .K67 1972

Library of Congress Cataloging in Publication Data

Korn, Granino Arthur, 1922–
Electronic analog and hybrid computers.

Includes bibliographies.
1. Electronic analog computers. 2. Hybrid computers. I. Korn, Theresa M., joint author. II. Title.
TK7888.K67 1972 621.3819'57 70-170174
ISBN 0-07-035363-8

Copyright © 1972, 1964 by McGraw-Hill, Inc. All Rights Reserved. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

1234567890 MAMM 765432

The editors for this book were Tyler G. Hicks and Frank Purcell, and its production was supervised by Stephen J. Boldish. It was set in Times New Roman 327. It was printed and bound by The Maple Press Company.

To the memory of Dean Roland George Dwight Richardson
of Brown University, who has contributed so much to the development
of applied mathematics in the United States,
this book is respectfully dedicated.

190913

Digitized by the Internet Archive
in 2019 with funding from
Kahle/Austin Foundation

https://archive.org/details/electronicanalog0000korn_m6b3

CONTENTS

Preface xv

Chapter 1. A Review of Electronic Analog Computation 1

Survey

1-1. Computers, Simulation, and Systems 1

Electronic Analog Computers

1-2. Machine Variables and Computing Elements 1

1-3. The Computer System 3

1-4. Operation of Electronic Differential-equation Solvers 5

1-5. Programming Differential Equations 5

1-6. Problem Scaling 8

(a) Amplitude Scaling 8

(b) Time Scaling 9

Analog-computer Representation of Linear Transfer Operators

1-7. Problem Statement 10

1-8. Differential-analyzer Techniques 10

1-9. The Impulse Response 12

1-10. Operational-amplifier Techniques 13

Some Programming Tricks

1-11. How to Avoid Division 16

1-12. Proper Use of Function Generators 21

1-13. Generalized Integration 22

References and Bibliography

Chapter 2. Iterative Analog Computation and Hybrid Analog-digital Computers 24

Iterative Analog Computation

2-1. A Critical Appraisal of Analog Computation 24

2-2. Evolution of Iterative Analog Computation 25

(a) How to Benefit from Analog-computer Speed 25

(b) Iterative-differential-analyzer Operation 27

(c) Analog Memory: Track-hold Pairs 30

(d) The INITIAL RESET Mode 33

Track-hold Operations with Sampled Data and Simulation of Digital Computers

2-3. Sampled-data Sequences and Solution of Difference Equations . . . 34

2-4. Linear Operations on Sampled Data 36

Hybrid Analog-digital Computation

2-5. Different Roles of Digital Computers in Hybrid Computation . . . 37

References and Bibliography

Chapter 3. Computer Accuracy. Design of Networks, Coefficient Potentiometers, and Operational Amplifiers 42

Computer Accuracy and Other Design Specifications

3-1. A Discussion of Analog-computer Accuracy 42
3-2. Component Accuracy: Static and Dynamic Errors 43
  (a) Static Accuracy, Step Response, and Small-signal Frequency Response 43
  (b) Phase Errors 44
  (c) Rate-limit Specifications and Switch-timing Errors 45
3-3. Other Design Specifications 46

Coefficient-setting Potentiometers

3-4. Specifications and Construction 47
3-5. Loading Considerations and Potentiometer Setting 50
3-6. Dynamic Errors and Capacitor Compensation 52

Computing Resistors and Capacitors. Parasitic-impedance Effects

3-7. Computing Resistors 56
  (a) Resistance Stability. Wirewound Computing Resistors 56
  (b) Parasitic Inductance and Capacitance. Film Resistors 57
3-8. Computing Capacitors: Stability 58
3-9. Capacitor Leakage and Dielectric Absorption 59
  (a) Resistive Leakage 59
  (b) Dielectric Absorption. Capacitor Models 59
  (c) Effects of Dielectric Absorption 61

The Parallel-feedback Operational Amplifier

3-10. Introduction 62
3-11. Operational-amplifier Performance with Finite Gain and Loading . . 62
3-12. Error Calculations 64
3-13. Effect of Amplifier Internal Impedance and Loading 65
3-14. Operational Amplifiers with Two-terminal Computing Impedances. Input-source Loading 66
3-15. Effects of D-c Offsets and Noise 67
3-16. Effects of Gain Changes and Distortion 68
3-17. Summary of Error Analysis 68

Design of Phase Inverters, Summing Amplifiers, and Integrators

3-18. Introduction 68
3-19. Design of Phase Inverters and Summing Amplifiers 68
  (a) Static Accuracy 68
  (b) Dynamic Accuracy and Phase-equalizing Capacitors 69
3-20. Electronic Integrators: Effects of Finite Gain, Loading, and Parasitic Impedances on the Integrator Frequency Response 72
3-21. Electronic Integrators: Errors Due to Calibration, D-c Offset, and Noise 76

Errors in Solutions of Linear Differential Equations with Constant Coefficients. System-design Considerations

3-22. Introduction 77
3-23. First-order Mode-error Calculations 78
3-24. Errors in Sinusoidal Modes. Circle Test and System Design . . . 80
  (a) The Circle Test 80
  (b) Examples and System-design Considerations 82

Differential and Non-inverting Operational Amplifiers

3-25. Introduction 86

3-26. Differential and Common-mode Amplification 86
3-27. Differential and Non-inverting Operational Amplifiers 88
3-28. Important Special Cases 90
3-29. Effects of D-c Offset and Noise 93
3-30. Input Impedances 93

D-c Amplifier Specifications, Design, and Tests

3-31. Survey 95
3-32. Drift Specifications 95
3-33. Low-drift Amplifier Design 96
3-34. Operational-amplifier Stability. Bode-plot Analysis of Stability and Accuracy 99
3-35. Feedforward Circuits 102
3-36. Chopper-stabilized D-c Amplifiers 104
3-37. Balancing and Test Circuits 107
3-38. Practical Frequency-response Measurements 110

References and Bibliography

Chapter 4. Nonlinear Operations: Limiters, Comparators, and Function Generators 114

Introduction

4-1. Function Generation and Switching 114

Simple Limiter and Selector Circuits

4-2. Junction-diode Characteristics and Diode Models 115
4-3. The Basic Diode Limiter Circuits 117
4-4. Low-frequency Analysis of Diode Circuits 117
4-5. High-frequency Performance of Diode Circuits 121
4-6. Diode-bridge Limiters 123
4-7. Temperature Compensation of Breakpoint Drift 125
4-8. Transistor Limiters 125

Analog Comparators

4-9. Comparator Circuits 127
  (a) Simple Operational-amplifier Comparators and Sign Multipliers . 127
  (b) Design of Improved Analog Comparators 128

Precision Limiters and Selectors

4-10. Precision Limiters, and Some Related Circuits 131
4-11. Accurate Selectors, Extreme-value-holding Circuits, and Peak Detectors . 137

Bistable Circuits and Special Signal Generators

4-12. Multivibrator Circuits 137
4-13. Accurate Sine-wave Oscillators 141

Electronic Function Generators

4-14. Introduction 144
4-15. Diode Function Generators 145
  (a) Special-purpose Function Generators. Smoothing Oscillators . . 145
  (b) General-purpose Function Generators 146
  (c) Multiple-function Generation 148
  (d) Function Setting and Storage 148
4-16. Logarithm-function Generators 150
4-17. Miscellaneous Circuits 152
  (a) General Remarks 152

(b) Varistor Function Generators 153
(c) Cathode-ray Function Generators 154

Functions of Two or More Variables

4-18. Introduction. Interpolation Schemes 155
4-19. Diode Function Generators 156
(a) Simple Limiter Circuits 156
(b) Use of General-purpose Diode Function Generators 156
(c) Special Diode Function Generators 158

References and Bibliography

Chapter 5. Switching, Conversion, Sampling, and Mode Control 163

Analog Switching Circuits

5-1. Applications and Desirable Specifications 163
5-2. Diode Switches 166
5-3. Bipolar-transistor Switches 168
(a) Static Characteristics 168
(b) The Inverted Transistor as a Switch 170
(c) Practical Switching Circuits 172
5-4. Field-effect-transistor Switches 174
(a) Junction Field-effect Transistors (JFETs) 174
(b) Metalized-oxide Semiconductor Field-effect Transistors (MOSFETs) 174
(c) Practical Switching Circuits 175
(d) Comparison of JFET, MOSFET, and Bipolar-transistor Switches . 177
5-5. Some Improved Switching Circuits 179

Digital-to-analog Converters

5-6. Basic Conversion Circuits. Representation and Conversion of Negative Numbers 180
5-7. Voltage-ladder Digital-to-analog Converters 182
5-8. Current-switching Circuits 184
5-9. Errors and Specifications 187

Electronic Analog-to-digital Converters

5-10. Introduction 191
5-10. Specifications 191
5-12. Ramp/comparator and Integrating Converters 192
(a) Ramp/comparator Converters 192
(b) Integrating Converters 193
5-13. Feedback-type Analog-to-digital Converters 195
(a) General Principle 195
(b) "Continuous" Converters 196
(c) Successive-approximation Converters 196
5-14. Converters Employing Multiple Comparators 199
5-15. Discussion. Accuracies and Conversion Rates 199

Integrator Mode Control and Sample-hold Circuits

5-16. Electronic Mode-control Switching 200
5-17. Track-hold Circuits (Sample-hold Circuits) 202
(a) Basic Track-hold Circuit. Performance Specifications 202
(b) Improved Track-hold Circuits 206
(c) Switched Integrators as Track-hold Circuits 206
5-18. Switched Operational Amplifiers 207

References and Bibliography

Chapter 6. Electronic Multipliers, Dividers, and Resolvers 212

Introduction
6-1. Survey 212
Quarter-square Multipliers and Related Circuits
6-2. Diode Quarter-square Multipliers 213
6-3. High-accuracy Squaring Circuits 216
6-4. Triangle-wave-averaging Multipliers 219
Multiplication through Voltage-Controlled Gain
6-5. Multiplication with Positive and Negative Inputs 221
6-6. Transistor Transconductance Multipliers 222
6-7. Feedback Control of Multiplier Gain 224
6-8. Examples of Feedback Multipliers 227
(a) Servo and Analog-digital Multipliers 227
(b) Photoresistor and Heat-transfer Multipliers 228
(c) Magnetoresistance and Hall-effect Multipliers 228
(d) Modulation-type Multipliers 229
Time-Division Multipliers
6-9. Principle of Operation 229
6-10. Externally Excited Time-division Multipliers 230
6-11. Self-excited Time-division Multipliers 231
(a) Basic Four-quadrant Multiplier 231
(b) A Modified Circuit 233
6-12. Static vs. Dynamic Accuracy in Multipliers Employing Ripple Filters. Ripple-filter Design 234
Some Improved Multiplication Schemes
6-13. High-frequency Correction of a Time-division Multiplier 236
6-14. Hybrid Analog-digital Multipliers Employing Coarse and Fine Channels 237
Electronic Multipliers: Testing and Division Circuits
6-15. Multiplier Adjustment and Testing 239
6-16. Electronic Division 240
Vector Resolution and Composition
6-17. Vector Resolution and Composition 242
6-18. Resolver Servos Using Sine-cosine Potentiometers 243
6-19. Induction Resolvers 244
6-20. Electronic Resolvers Combining Function Generators and Multipliers . 245
6-21. Quadrant Switching and Rate Resolvers 245
6-22. Inverse Resolvers (POLAR Resolver Mode) 249
6-23. Use of Sampling and Comparator Circuits 250
6-24. Implicit-computation Schemes 252
References and Bibliography

Chapter 7. Computer Systems and Interface Design 259

Special-purpose Systems
7-1. Special-purpose Computer Systems and Subsystems 259
General-purpose Analog/hybrid Computers
7-2. Typical Installations and General Design Principles 259

(a) Small Tabletop Computers 259
(b) Medium-size and Large Installations 262
(c) Power Supplies 264
7-3. Logic and Control Systems 264
(a) Mode Control and Patched Digital Logic 264
(b) Timing Circuits 265
(c) Patched Digital Logic 268
7-4. Patching and other Programming Systems 268
(a) Analog-computer Patchbays 268
(b) Setting Coefficients and Functions 271
(c) Automatic Patching and Time-shared Analog Computation . . . 271
7-5. Readout Systems 275
(a) Analog-computer Readout 275
(b) Strip-chart Recorders 275
(c) Servo Plotting Boards and Curve Followers 278
(d) Amplifier-readout Switching 278
7-6. Calibration and Program Checks 279
(a) Calibration and Amplifier-balance Checks 279
(b) Overload-alarm Systems and Minimum-excursion Indicators . . . 279
(c) Static Checks, Rate Tests, and Time-scale Check 280
7-7. Analog-computer Accuracy Checks and Maintenance Procedures . . . 281

Digital Computer and Interface

7-8. Survey: Digital Computer and Linkage Requirements 281
7-9. Programmed I/O Operations 284
(a) The Party-line I/O Bus 284
(b) Program-controlled Device Selection and Operation 285
(c) Data Transfers and other Device Operations 286
7-10. Sense-line Operation 288
7-11. Interrupt-system Operation 289
7-12. Direct Memory Access and Block Transfers 291
(a) Cycle Stealing 291
(b) Automatic Block Transfers 293

Noise, Ground Systems, and Interconnections

7-13. Noise and Ground Systems 294
7-14. Digital-system Noise. Interconnection Problems 296

References and Bibliography

Chapter 8. Special Computing Techniques 301

Introduction and Survey

8-1. Requirement for Advanced Computing Techniques 301

Gauging and Improving Computer Accuracy: Perturbations, Parameter Sensitivity, Resubstitution, and Automatic Scaling

8-2. Perturbations and Linearization 302
8-3. Example: Earth-satellite Orbit and Reentry Computation 303
8-4. Parameter-influence Coefficients and Parameter-sensitivity Studies . . 305
8-5. Resubstitution Checks 306
8-6. Special Scaling Problems and Automatic Scaling 306

Continuous Optimization, Equation Solvers, Constraints, and Model Matching

8-7. Maxima and Minima 307
(a) Introduction 307
(b) Automatic Optimization: Continuous Steepest Descent 309

8-8. Solution of Equations . . . 310
(a) Algebraic and Transcendental Equations . . . 310
(b) Simultaneous Linear Equations . . . 311
(c) Hybrid Iteration . . . 312
8-9. Steepest Descent and Dynamical Constraints . . . 312
(a) Dynamical Constraints . . . 312
(b) Steepest-descent Methods for Improving Differential-analyzer Accuracy . . . 314
8-10. Linear Programming and Related Problems . . . 317
8-11. Optimization by Parameter Perturbation . . . 320
8-12. Continuous Optimization and Model Matching . . . 321

Iterative Optimization

8-13. Parameter Optimization of Functionals. Examples . . . 323
8-14. Iterative-differential-analyzer Techniques . . . 326
(a) Introduction . . . 326
(b) Automatic Iteration . . . 327
(c) Use of Parameter-influence Coefficients. Kokotovic Systems . . . 328
(d) Computed-gradient Methods . . . 331
(e) Termination of the Iteration. Step-size Determination . . . 331
(f) Other Optimization Routines . . . 335
8-15. Optimization by Creeping Random Search. Global Searches and Constraints . . . 335
8-16. True Functional Optimization . . . 339
(a) Solution by Calculus of Variations and Maximum Principle . . . 339
(b) Example: Simple Bang-bang Time-optimal Control . . . 342
(c) Direct Functional Optimization . . . 343

Solution of Partial Differential Equations

8-17. Survey . . . 344

Simulation of Special Control-system Nonlinearities

8-18. Saturation, Limit Stops, and Backlash . . . 344
8-19. Simulation of Static and Coulomb Friction . . . 350
8-20. Other Common Nonlinearities . . . 351

References and Bibliography

Chapter 9. Hybrid-computer Errors, Special Techniques, and Software Requirements . . . 361

Introduction

9-1. Survey . . . 361

Errors in Hybrid Computation

9-2. Hybrid-computer Error Sources . . . 361
9-3. Sampling and Data Reconstruction . . . 363
9-4. Errors due to Processing Delays in Combined Analog-digital Simulation . . . 365
9-5. Reducing Sampled-data Errors . . . 367
(a) General Remarks . . . 367
(b) Intermittent-HOLD Operation . . . 367
(c) Digital-computer Prediction . . . 368
(d) Analog-computer Prediction . . . 368
(e) Perturbation and Hybrid-code Methods . . . 369
9-6. Conclusions . . . 370

Hybrid-computer Function Storage and Function Generation

9-7. Function Storage. Fixed and Variable Time Delays 371
9-8. Hybrid-computer Techniques for Function Generation 372
(a) Introduction 372
(b) Digital Table-lookup and Digital Interpolation 373
(c) Digital Table-lookup and Hybrid-code Interpolation Using Uniformly Spaced Breakpoints 374
(d) Digital Table-lookup and Analog Interpolation 375
9-9. Functions of Two or More Variables 375
9-10. Discussion 377

Programs and Software for Hybrid Analog-digital Computation

9-11. Software Requirements 378
9-12. Library Routines and their use in Programs and with Interpreters . . 379
9-13. Typical "Main Programs" 380
(a) Data Processing 380
(b) Digitally Controlled/Iterative Analog Computation 380
(c) Closed-loop Combined Analog-digital Simulation 381
(d) Discussion. DMA Operations and Hardware/Software Tradeoffs . 381
(e) Interrupt Servicing and Reentrancy 383
9-14. Operating Systems 383
9-15. Use of Simulation Languages 384
9-16. Hybrid-computer Diagnostic Programs 386
9-17. Time Sharing 387

References and Bibliography

Appendix A: Operational-amplifier Design Data 391

Appendix B: Operational-amplifier Circuits for Time-delay Simulation . . 399

Appendix C: Logic Circuits and Symbols 405

Author Index 409

Subject Index 415

PREFACE

Hybrid analog-digital computation has added digital memory, automatic program control, and powerful digital operations to the familiar analog computer. At the same time, new discrete and integrated-circuit amplifiers, multipliers, and switches have increased computing speed and reliability by two orders of magnitude. Month-long optimization and statistical simulation studies have been contracted into days and hours. Nevertheless, simulation with general-purpose hybrid computers forms only a small part of the total thrust of hybrid computation.

The real impact of analog-digital circuits and techniques applies to control, instrumentation, and communication systems. In many such systems, one must interface large and small digital computers to analog equipment in the real world. Thus, an entire new industry supplies constantly improving hybrid analog-digital computing devices for handling live instrumentation and for controlling data in industry, communications, weapons, and in space.

This is not a book on routine analog-computer simulation; there are, surely, enough excellent texts on this subject. We believe that this volume is unique in its emphasis on presenting *hardware information for designers of hybrid analog-digital equipment.* We also introduce some *improved computing techniques made possible by the new, ultra-fast hybrid computers,* especially in connection with optimization.

This completely revised edition, which again includes much previously unpublished information, will, we hope, be useful to research workers and designers. The subject matter should be readily accessible to seniors in electrical engineering or physics. Lecture notes based on the same material have served as a text for a series of special senior-elective and graduate courses involving development projects instead of term examinations. These courses have employed small hybrid computers less for differential-equation solving than for breadboarding analog-digital instrumentation.*

---

* G. A. Korn, Analog/hybrid Computers in Engineering Education, *JASEE,* April 1968; *Engineering Experiment Station Report No. 16,* University of Arizona, is a laboratory manual.

With well over 250 computer circuits, block diagrams, and photographs, this is truly a picture book, and the reader will be able to form a fair preliminary idea of the text material by first looking at the illustrations. Circuit diagrams are presented in related groups to permit convenient comparisons. We have also attempted to give the reader convenient access to the ever-growing computer literature by listing over 730 bibliographical references, by topics and subtopics, at the end of related chapters; a special author index provides alphabetical access to this bibliography.

Chapter 1 very briefly reviews *the concepts of "classical" analog computation* and presents a time-tested method for *analog/hybrid-computer scaling.* Computer implementation of *linear transfer operators,* which is becoming more important for active-filter synthesis with integrated circuits, is described next; our treatment includes computation of impulse responses for use in more advanced calculations. *General-purpose operational amplifiers,* now widely and inexpensively available to integrated-circuit form, are also introduced in this first chapter, with emphasis on control and instrumentation applications.

Chapter 2 introduces *the basic methods of hybrid-computer simulation,* viz., digitally controlled/iterative analog computation and combined analog-digital simulation. The different roles of digital computers in these applications are discussed. Two sections describe *methods for solving difference equations with track-hold circuits.*

Chapters 3 through 6, containing much new material, deal with the actual *design and specifications of modern analog hybrid computing elements.* Chapter 3 presents detailed methods of *error analysis for linear computing networks and operational amplifiers* and discusses *errors in computer solutions of linear differential equations.* Thirteen new sections describe the *specification and design of operational amplifiers,* including the more recent *circuits with differential inputs.*

Chapter 4 deals with *diode and transistor limiters,* modern fast *comparators,* and accurate *precision-limiter and selector circuits.* The chapter ends with a discussion of versatile *signal generators* built with operational amplifiers and of *general-purpose function generators.*

The next class of basic hybrid computing elements, namely *analog switching circuits, analog-to-digital converters* and *digital-to-analog converters,* are discussed in Chapter 5. Specifications, design, and error analysis of diode-, transistor-, and field-effect-transistor switches are included, as are switching circuits for *integrator mode control* and *sample-hold* operations. A comprehensive discussion of *analog-digital-analog conversion* is presented next.

Chapter 6 introduces *analog multipliers, dividers, and resolvers,* with many practical circuits.

Our new Chapter 7 is concerned with the design of *complete analog-*

digital computer systems including *general-purpose hybrid computers* and *automatic patching systems.* Five sections describe the operation of *digital-computer interfaces* to external devices, with party-line input/output buses, device selection, sense lines, interrupts, and direct-memory-access data channels. The chapter ends with sections on the important topics of *ground systems* and *digital-noise problems.*

Chapter 8 introduces some of the new *mathematical techniques* opened up by analog/hybrid computation. The chapter begins with accurate *perturbation methods* for space-vehicle trajectory computation and *steepest-descent methods* for continuous optimization and improvement of computing accuracy. A major portion of Chapter 8 is devoted to methods for *automatic parameter optimization,* including the use of *sensitivity techniques.* New sections deal with the *Kokotovic linear-system technique,* and the increasingly important method of *creeping random search.* Both *parameter optimization* and *true functional optimization* are discussed.

Chapter 9 is entirely new. The chapter first discusses the *errors inherent in the sample of data operation of hybrid-computer systems.* Circuits for hybrid-computer *function storage* (including *simulation of fixed and variable time-delays*) and *function generation* are next described. The seven last sections describe *software requirements* for hybrid computation and include discussions of utility library routines, hybrid-computer interpreters, main-program structure, interrupt servicing, reentrancy, and hybrid-computer operating systems. The chapter ends with a brief discussion of the use of simulation languages, hybrid-computer diagnostic programs, and time-sharing problems.

Two *appendixes* present operational amplifier design data and the logic circuits and symbols employed in the rest of the volume.

We originally intended to include two additional chapters describing analog/hybrid computer applications to random-process simulation and measurements. In particular, Monte-Carlo-type simulation of control, communication, and detection systems is a striking application of the modern high-speed iterative differential analyzer. Computer techniques for noise generation and statistical measurements are widely useful. For better or for worse, however, our account of these techniques (which have been the subject of much of our research at the University of Arizona) proved to be too extensive for inclusion in this book. Since, too, this material is not directly related to other topics treated in this volume and also involves some additional mathematics, it has been published as a separate monograph.* The expanding and promising field of hybrid-computer techniques for solving partial differential equations, too, could not be done justice in a brief review and deserves a separate book.

---

* G. A. Korn, *Random-process Simulation and Measurements,* McGraw-Hill, New York, 1966.

The writers are grateful to the University of Arizona and especially to Dr. R. M. Mattson for encouragement of our hybrid-computer research, and to the National Science Foundation, the Air Force Office of Scientific Research, the National Aeronautics and Space Administration for their support of the APE, ASTRAC II, and LOCUST hybrid-computer projects. We would like to thank the many students whose term papers and thesis work contributed to our research, and especially to our associates in the University's Computer Science Research Laboratory:

H. Amerine
M. Andrews
H. Aus
J. Belt
T. Brubaker
D. Chinnock
B. Conant
H. Eckes
D. Eddington
C. Foiles
J. Ferguson
J. Goltz
R. Gonzalez
R. Hampton
H. Handler
J. Hartmann
J. Jensen
H. Koerner
H. Kosako
T. Liebert
R. Mangels
R. Maybach
B. Mitchell
J. Moore
M. Naka
E. O'Grady
C. Pracht
J. Puls
S. Simons
A. Trevor
J. Wait
R. Whigham
R. White
C. Wiatrowski
J. Wilkins
A. Yagi

Last, but not least, we are, once again, grateful to our friends in the computer industry for contributing their know-how in the form of reports, circuits, photographs, and discussions. We are, in particular, indebted to the following individuals and organizations:

Applied Dynamics, Inc. Ann Arbor, Michigan	Mr. G. Graber
Burr-Brown Research Corp. Tucson, Arizona	Mr. H. Eckes
Electronic Associates, Inc. Long Branch, New Jersey	Mr. D. Darms Mr. R. Doelger
Hybrid Systems, Inc. Houston, Texas	Mr. H. Durrett
Simulation Councils, Inc.	Mr. J. McLeod Mr. S. Rogers

Granino A. Korn
Theresa M. Korn

ELECTRONIC ANALOG
AND HYBRID COMPUTERS

CHAPTER 1

A REVIEW OF ELECTRONIC ANALOG COMPUTATION

SURVEY

1-1. Computers, Simulation, and Systems. *Computers* are physical systems designed to implement the idealized relationships of mathematical models. Computers are conventionally classified as (1) *analog computers* when the machine variables representing mathematical quantities are considered to vary continuously (voltages, shaft rotations) and (2) *digital computers* when they are discrete-state systems. *Hybrid analog-digital computation* employs a variety of techniques to combine high analog computing speed with accurate digital operations, decision making, and memory.

Computers—analog, digital, or hybrid—have two principal applications. They may set up mathematical relations to produce desired unknown or otherwise desired quantities and/or functions (*problem solving*); in particular, *computer simulation* substitutes a computer-implemented model for real system components in *experiments* conducted for purposes of design, training, or partial system tests. The second (and at least as important) application of computing devices is as *system components* used for instrumentation, control, and display: here, *the computer forces at least a portion of a reluctant physical system to act like a mathematical model and thus recreates the system nearer to the heart's desire.*

ELECTRONIC ANALOG COMPUTERS

1-2. Machine Variables and Computing Elements. *Electronic analog computers* (Fig. 1-1) most frequently represent each problem variable by a corresponding voltage between ±100 or ±10 volts on a convenient

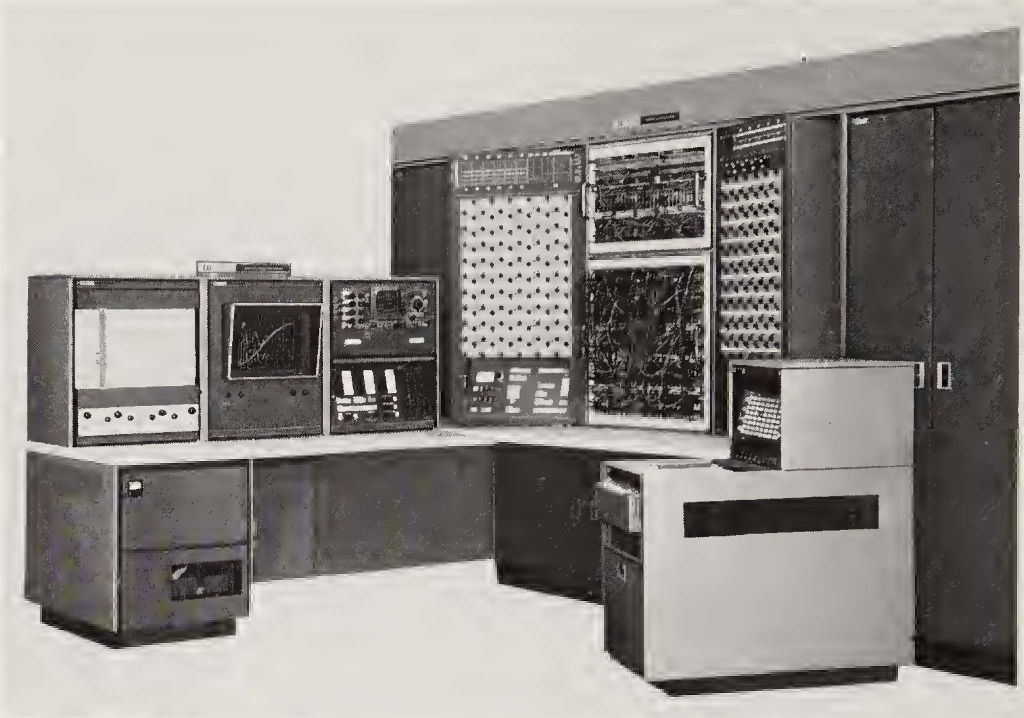


Fig. 1-1*a*. A large analog/hybrid computer console with up to 250 ± 100-volt amplifiers. Such installations, which may comprise several similar consoles, are usually controlled by a medium-sized digital computer. Solutions can be read with elaborate and convenient display/plotter facilities on the left and with the multichannel strip-chart recorder on the right. (*Electronic Associates, Inc., Type* 8800 *computer.*)

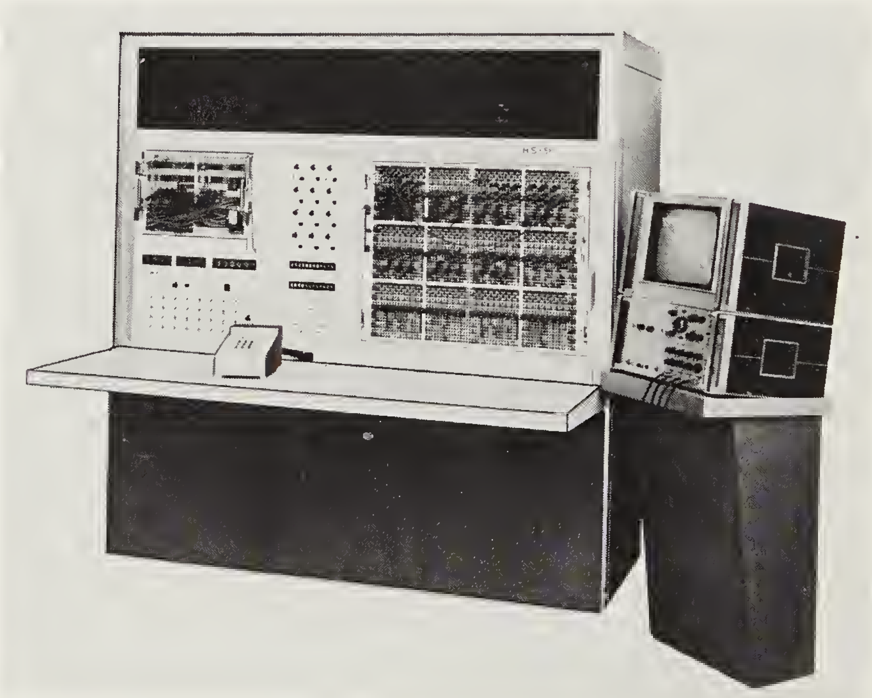


Fig. 1-1*b*. A moderately fast medium-sized ±100-volt computer, the Hybrid Systems SS-50 has phase inverters accurate to within 0.1 percent up to 1 KHz. The machine is also sold with a 16-bit minicomputer built directly into the interface. (*Hybrid Systems, Inc.*)

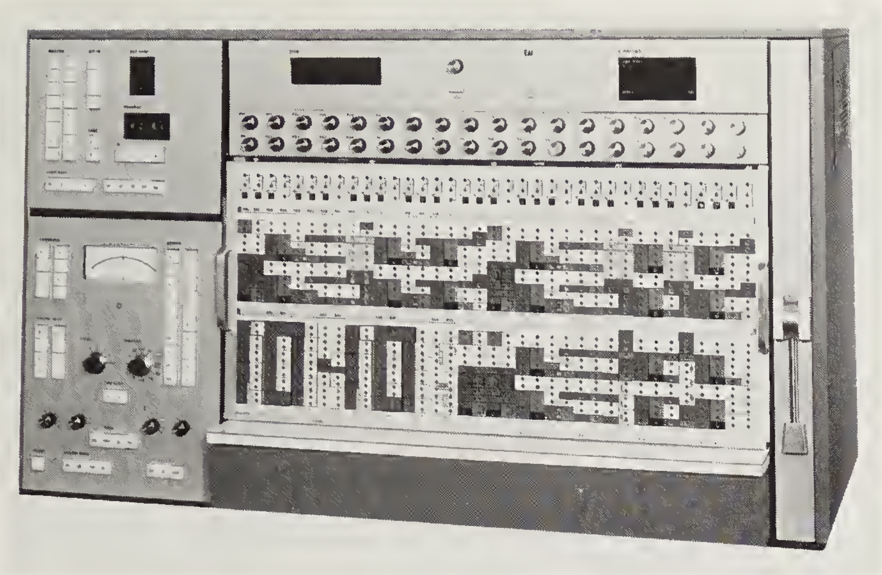


Fig. 1-1c. EAI 380 is a smaller tabletop machine with up to 54 ± 10-volt amplifiers. Diode-function-generator units are in a drawer underneath the unshielded patchbay, which accommodates both analog computing elements and digital logic. (*Electronic Associates, Inc.*)

scale (*problem scaling,* Sec. 1-6). These *machine variables* are made to obey mathematical relations corresponding to those of the given problem. Electronic analog computers establish more or less complex mathematical relationships between machine variables (voltages) $X_0, X_1, X_2, \ldots$, say

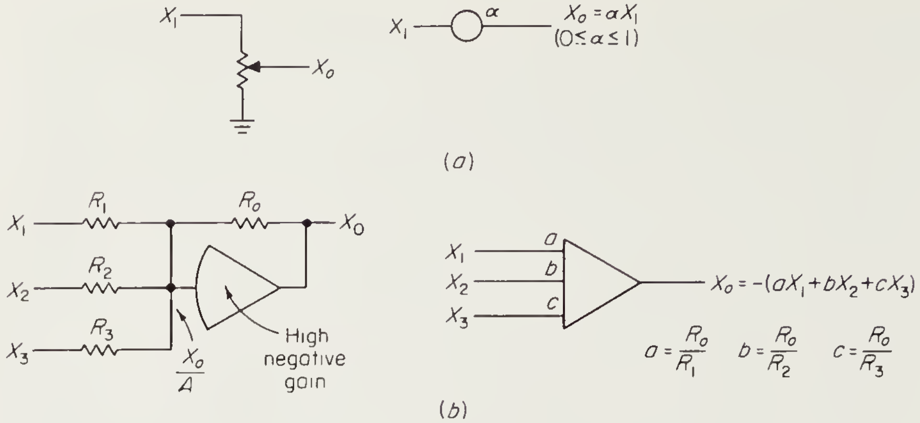
$$X_0 = 0.02X_1X_2 - 100 \sin 3X_2 = F(X_1,X_2) \tag{1-1}$$

by combining computing elements establishing elementary mathematical operations, usually:

Multiplication of a machine variable (voltage) by a constant coefficient
Addition of two or more machine variables
Multiplication of two machine variables
Generation of functions of machine variables

(Fig. 1-2). In addition, analog differential-equation solvers (differential analyzers, Sec. 1-4) employ electronic *integrators* capable of producing the integral of a voltage $x(\tau)$ with respect to the computer time τ, which serves as the independent variable (Fig. 1-3). Conventional electronic analog computers employ separate analog computing elements for each of the operations required by a simulation or data-processing problem.

1-3. The Computer System. Besides a set of operational amplifiers and other computing elements and their power supplies, an electronic



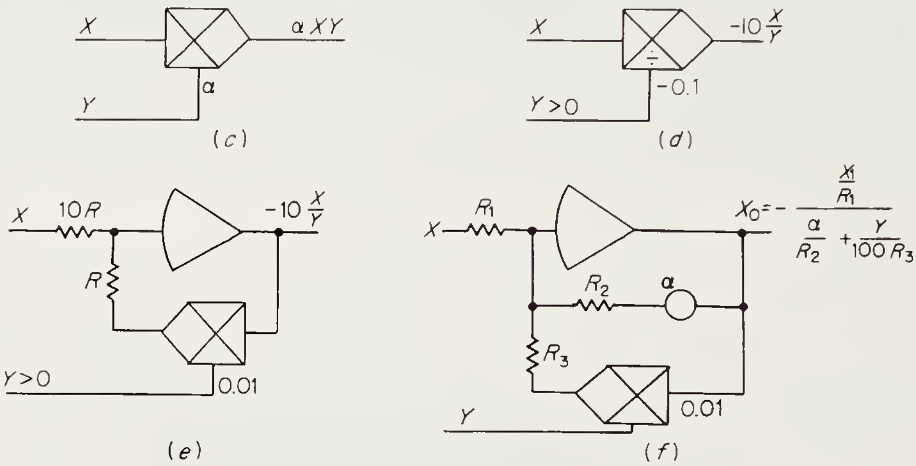


Fig. 1-2*a*, *b*. Coefficient-setting potentiometer (*a*), and summer/phase inverter (*b*), with block-diagram symbols. High-gain degenerative feedback makes the operational-amplifier input voltage in (*b*) very small, so that current summation yields

$$X_0 = -R_0\left(\frac{X_1}{R_1} + \frac{X_2}{R_2} + \frac{X_3}{R_3}\right)$$

(see Sec. 3-11 for detailed circuit analysis).

Fig. 1-2*c* to *f*. Electronic multiplier (*c*), divider (*d*), and feedback circuits implementing division with multipliers (*e, f*) (see also Secs. 6-16 and 8-2).

analog-computer system will comprise:

1. *Interconnections* between computing elements to implement the analog-computer program. Interconnections may involve fixed wiring and switches in special-purpose machines or *patching* connections in general-purpose computers.
2. *Control switches or relays* to place initial conditions on integrators (RESET *condition*) and to start and terminate the computation (Fig. 1-3*b*).
3. *Readout devices*, such as oscilloscopes, strip-chart recorders, *xy* recorders, digital voltmeters, and printers.

4. *Reference power supplies* to supply accurate *reference voltages* (usually ±100 or ±10 volts) required as computing and initial-condition voltages.

1-4. Operation of Electronic Differential-equation Solvers. The electronic analog computer represents problem variables by voltages which vary as functions of the time τ (*machine time*, used as the independent variable). The *solution of a set of ordinary differential equations* proceeds as follows:

1. With the machine connected up to solve the given problem, the machine variables (voltages) are set to the correct initial conditions prescribed by the problem (RESET *mode*).
2. Computing elements, particularly integrators, are made operative and force the voltages in the machine to vary in a manner prescribed by the given differential equations (COMPUTE *or* OPERATE *mode*). The voltage variations with time are recorded and constitute solutions of the given problem.
3. The machine is reset to its initial conditions and is ready for the next run with changed coefficients, initial conditions, etc.

1-5. Programming Differential Equations. Figure 1-4*a* illustrates the solution of a first-order ordinary differential equation

$$\frac{dX}{d\tau} \equiv PX = -F(X,\tau) \tag{1-2}$$

by an analog integrator set up to integrate a voltage $-F(X,\tau)$ obtained by suitable combinations of coefficient potentiometers, summing amplifiers, multipliers, and function generators; here, X is a voltage (machine

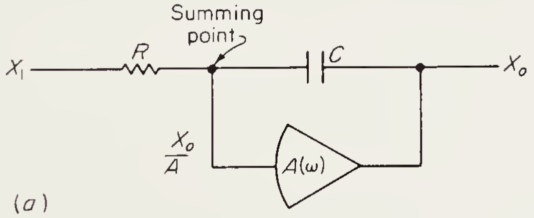


Fig. 1-3*a*. Operational-amplifier integrator. The high-gain feedback circuit reduces the summing-point voltage X_0/A to a very small value at frequencies $\omega/2\pi$ such that $A(\omega)|\,\omega RC \gg 1$, so that the summing-point node equation

$$\frac{1}{R}\left(X_1 - \frac{X_0}{A}\right) + \frac{1}{C}\int\left(X_0 - \frac{X_0}{A}\right)d\tau = 0$$

implies

$$X_0 = -\frac{1}{RC}\int X_0(\tau)\,d\tau = -\frac{1}{RC}\int_0^\tau X_0(\tau)\,d\tau + X_0(0)$$

where $X_0(0)$ is the initial voltage on the integrating capacitor (initial-condition voltage). The low-frequency amplifier gain $A(0)$ is real and negative (between -10^4 and -10^8).

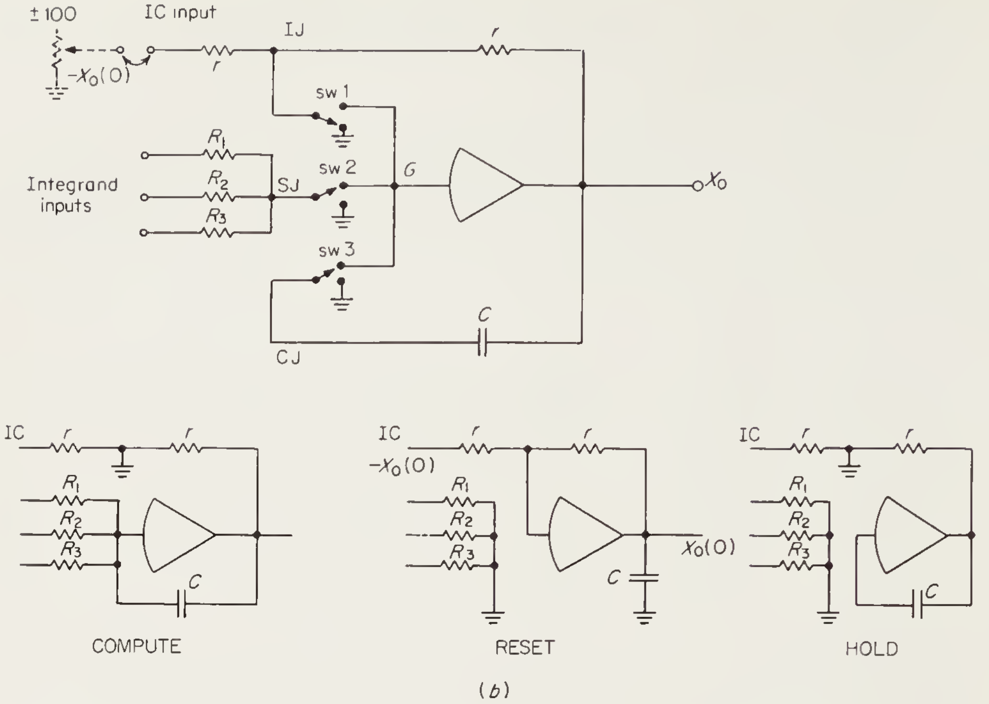


Fig. 1-3*b*. Integrator mode-control circuits. Before every computer run, the integrating capacitor must be charged to the correct initial value $X_0(0)$. In the COMPUTE mode, *SW*1 grounds *IJ*, while *SW*2 and *SW*3 establish the integrator input and feedback connections. In RESET:

1. *SW*1 connects the amplifier as a phase inverter for the initial-condition (IC) input $-X_0(0)$, so that the output becomes $X_0(0)$.
2. *SW*2 disconnects the integrand inputs and grounds the summing junction SJ so that the integrand sources see the same loads as in COMPUTE (this is important if these sources are coefficient-setting potentiometers or networks, Sec. 3-5).
3. *SW*3 grounds *CJ*, so that the amplifier charges the integrating capacitor to the correct initial voltage $X_0(0)$.

Note that the IC input voltage is *inverted*, like other operational-amplifier input voltages. The integrator of Fig. 1-3*b* has a third mode, the HOLD mode: *SW*2 again disconnects and grounds the summing junction *SJ*, but *SW*3 retains the capacitor feedback. The circuit is, then, an integrator with zero input and *holds* the output voltage $X_0(\tau)$ it had when going into the HOLD mode (analog memory, see also Sec. 2-3).

For accurate solution of differential equations, all integrators must switch from RESET to COMPUTE within a very short time interval (Sec. 3-2*c*). Similarly, an integrator must switch quickly from COMPUTE to HOLD if the retained output voltage is to be meaningful (Sec. 5-17). Early analog computers employed *relays* switched individually or by common COMPUTE, RESET, or HOLD buses (Sec. 7-3) for mode control, but faster computation requires *electronic* mode-control switching, which will be discussed in more detail in Sec. 5-16.

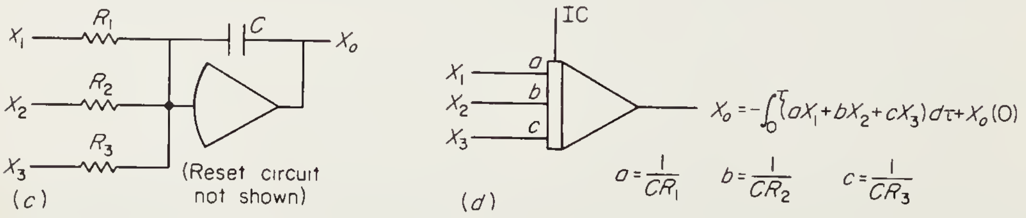


Fig. 1-3*c*, *d*. Summing integrator (*c*) and block-diagram symbol (*d*).

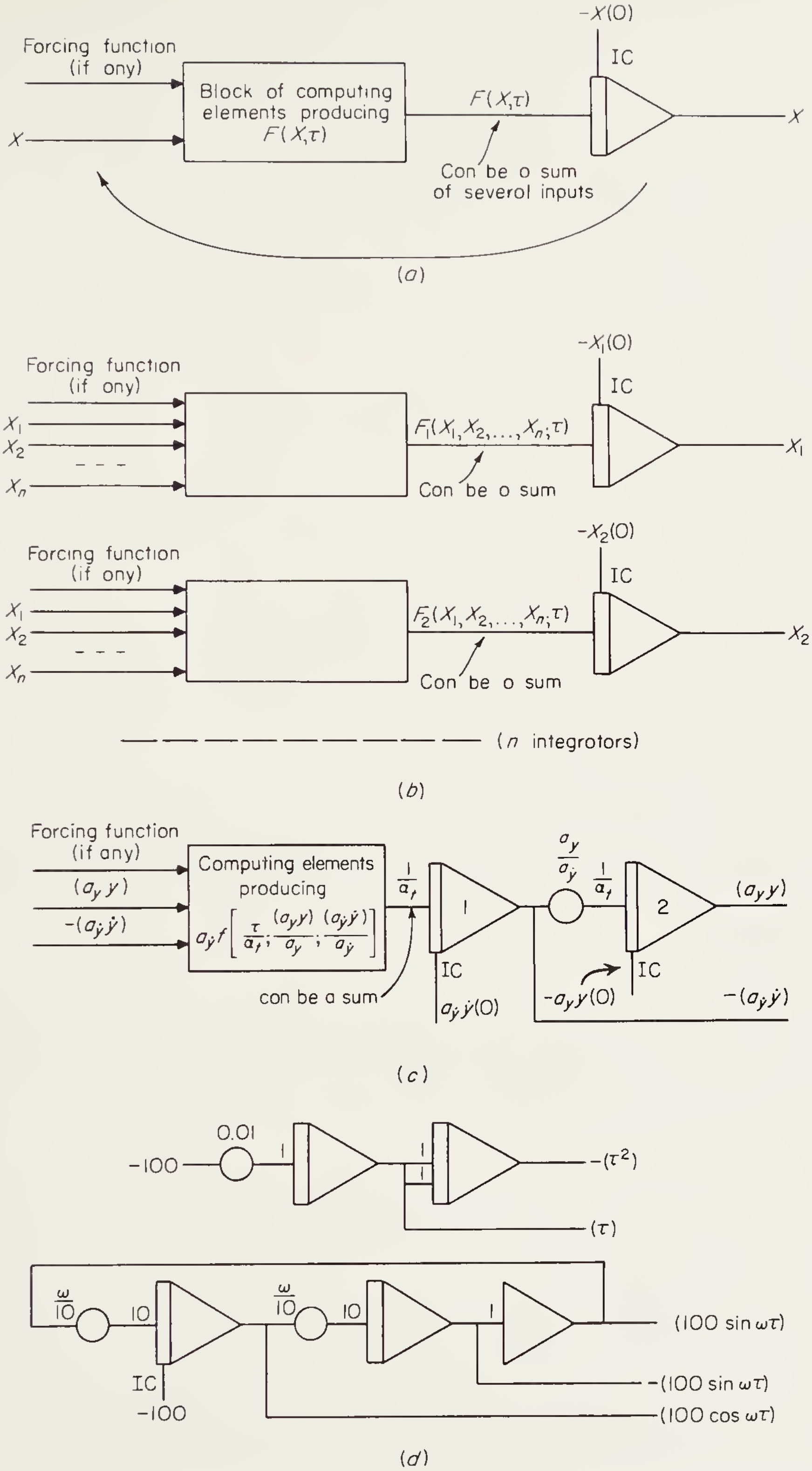


Fig. 1-4. Solution of a first-order differential equation (a), of n first-order equations (b), and scaled computer setup for a second-order equation (c). Figure 1-4d shows how forcing functions (input functions of the computer time τ) are obtained as solutions of suitable differential equations.

variable), and the independent variable τ is real time (computer time). Note that feedback of the voltage X around the integrator is involved (implicit computation, classical differential-analyzer technique), and that the integrator output starts with a specified *initial-condition voltage.*

In Fig. 1-4*b*, n integrators similarly solve n simultaneous first-order differential equations

$$\frac{dX_i}{d\tau} \equiv PX_i = -F_i(X_1,X_2, \ldots , X_n;\tau) \qquad (i = 1,2, \ldots ,n) \quad (1\text{-}3)$$

We shall transform higher-order differential equations into sets of first-order equations by introducing derivatives $\dot{x}, \ddot{x}, \ldots$ as new variables. In particular, a second-order differential equation (e.g., an equation of motion)

$$\frac{d^2x}{d\tau^2} = f\left(x, \frac{dx}{d\tau}\,;\tau\right) \quad (1\text{-}4)$$

is transformed into a pair of first-order equations

$$\frac{d\dot{x}}{d\tau} = f(x,\dot{x}\,;\tau) \qquad \frac{dx}{d\tau} = \dot{x} \quad (1\text{-}5)$$

(Fig. 1-4*c*).

Input functions of the computer time τ (*forcing functions*) can themselves be generated by solution of suitable differential equations (Fig. 1-4*d*), or special signal generators (e.g., sine- and square-wave generators) may be used.

1-6. Problem Scaling. (a) Amplitude Scaling. Analog computers represent problem variables $x,y, \ldots$ by machine variables (usually voltages, currents, or shaft rotations)

$$X = a_x x \qquad Y = a_y y \qquad \cdots \quad (1\text{-}6)$$

The dimensional *scale factors* $a_x, a_y, \ldots$ are chosen so that each machine variable $X, Y, \ldots$ is as large as possible without exceeding its maximum permissible excursion $X_{\max}, Y_{\max}, \ldots$ (e.g., 10, 50, or 100 volts) in absolute value.

To scale a given problem for an electronic analog computer capable of supplying ±100-volt machine variables, one may measure $X, Y, \ldots$ in 100-volt *machine units*, but *scaling in volts* is almost universally accepted for such computers. If we scale in volts, we must find new scale factors if we use a ±10-volt machine. This difficulty is avoided by scaling in terms of *percent machine units.* One percent machine unit is just 1 volt for the common ±100-volt analog computers. Scaling for ±10-volt computers is then identical to that for ±100-volt machines, although one percent machine unit is now 0.1 volt. We have

$$a_x \leq \frac{100}{|x|_{\max}} \frac{\text{volts}}{\text{problem unit}} \tag{1-7}$$

where $|x|_{\max}$ is the largest expected excursion of the problem variable x. For convenience, the numerical value of each scale factor is usually a round number of the form 10^n, 2×10^n, 0.25×10^n, 4×10^n, or 5×10^n.

To obtain the correct *machine equations* relating the voltages $(a_x x)$, $(a_y y)$, . . . , write

$$x \text{ as } \frac{1}{a_x}(a_x x) \qquad y \text{ as } \frac{1}{a_y}(a_y y) \qquad \cdots \tag{1-8}$$

in each given problem equation. Then set up the block diagram, leaving terms like $(a_x x)$, $(a_y y)$ intact; these terms will appear as voltages whose absolute values cannot exceed 100 volts or 100 percent of a machine unit. Note that computer setups for second-order differential equations (4), (5) *require separate scale factors for x and $\dot{x}$.* $|x|_{\max}$ and $|\dot{x}|_{\max}$ are estimated from physical considerations or by exploratory computer runs with small scale factors.

(b) Time Scaling. Electronic integrators and computing networks integrate machine variables *with respect to real time.* Therefore, *real time, or machine time τ, is the independent machine variable in an electronic differential analyzer.* The *time scale* is established by writing a transformation equation relating the machine time τ to the mathematical independent variable t,

$$t = \frac{1}{\alpha_t}\tau \tag{1-9}$$

The coefficient α_t is the *time-scale factor;* note that $\alpha_t > 1$ for a *slow* time scale where t sec of problem time are represented by a *longer* period τ of computer time. $\alpha_t < 1$ corresponds to a *fast* time scale.

The correct machine equations are obtained through the substitution

$$\frac{d}{dt} = \alpha_t \frac{d}{d\tau} \equiv \alpha_t P \tag{1-10}$$

When the time scale is changed, the voltage changes in the computer remain proportional to the corresponding changes of the mathematical variables, but the *rates* at which these voltages change are speeded up or slowed down to improve accuracy or increase the convenience of the computation. A simple way to increase (slow) the time scale is to increase all capacitors in the computer setup proportionately, and vice versa.

If the time t occurs *explicitly* in a given differential equation, t will be represented by a machine variable (voltage) $(a_t t)$ in addition to the machine τ; one must not confuse a_t with the time-scale factor α_t.

ANALOG-COMPUTER REPRESENTATION OF LINEAR TRANSFER OPERATORS

1-7. Problem Statement. The dynamical relationship between the input (stimulus) $x(t)$ and the output (response) $y(t)$ of many system components can be represented by a linear differential equation

$$\frac{d^n y}{dt^n} + a_{n-1}\frac{d^{n-1}}{dt^{n-1}} + \cdots + a_0 y = b_m \frac{d^m x}{dt^m} + b_{m-1}\frac{d^{m-1}x}{dt^{m-1}} + \cdots + b_0 x \qquad (m \leq n) \quad \text{(1-11}a\text{)}$$

with constant coefficients a_i, b_k, or in operator form

$$y(t) = H(p)x(t) = \frac{b_m p^m + b_{m-1}p^{m-1} + \cdots + b_0}{p^n + a_{n-1}p^{n-1} + \cdots + a_0}\, x(t) \qquad \left(m \leq n;\ p \equiv \frac{d}{dt}\right) \quad \text{(1-11}b\text{)}$$

1-8. Differential-analyzer Techniques. To set up Eq. (11) on a general-purpose electronic differential analyzer, we can:

1. Integrate Eq. (11*a*) *n* times to obtain

$$y = b_n x + \frac{1}{p}(b_{n-1}x - a_{n-1}y) + \frac{1}{p^2}(b_{n-2}x - a_{n-2}y) + \cdots + \frac{1}{p^n}(b_0 x - a_0 y) \quad \text{(1-12)}$$

which is implemented in Fig. 1-5*a*.

2. Rewrite Eq. (11*b*) as

$$y(t) = (b_n p^n + b_{n-1}p^{n-1} + \cdots + b_0)z(t) \quad \text{(1-13)}$$

where

$$(p^n + a_{n-1}p^{n-1} + \cdots + a_0)z(t) = x(t) \quad \text{(1-14)}$$

The computer setup of Fig. 1-5*b* solves Eq. (14) in the ordinary manner and generates $z, -pz, p^2 z, \ldots$. These quantities are combined in accordance with Eq. (13) to produce y.

High-order operators (11) are usually factored into lower-order terms. In most practical applications, initial integrator outputs in Fig. 1-5 can be equal to zero. Scaling in these computer setups may require a trial-and-error procedure. Computer setups for operations of the form (11) with *time-variable* coefficients a_i, b_k are discussed in Ref. 19.

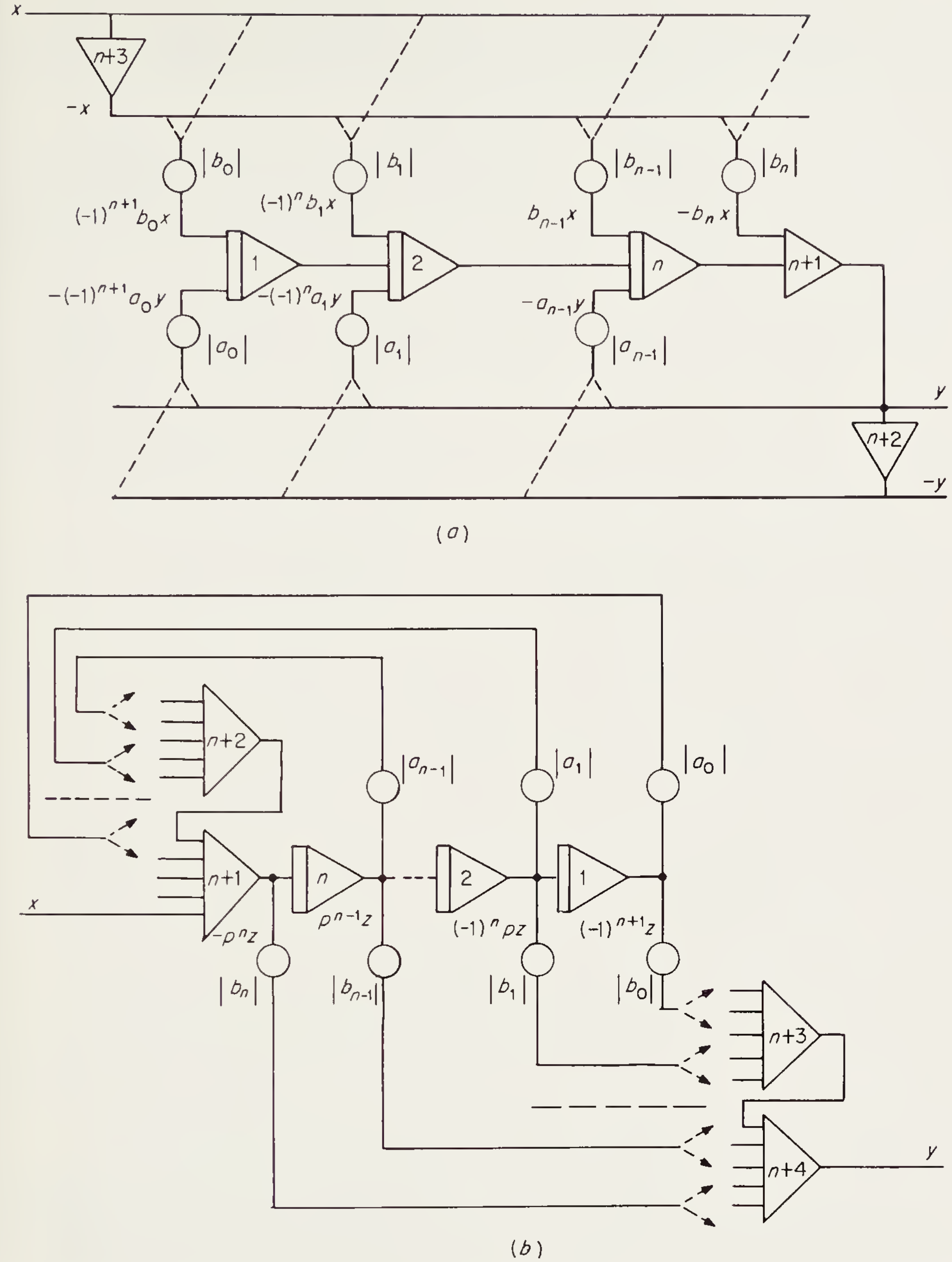


Fig. 1-5. Transfer-operator setups by Method 1 (*a*), and by Method 2 (*b*). Each setup procedure requires n integrators; in general, Method 1 requires three summer/phase inverters, and Method 2 requires four summer/phase inverters. Each coefficient a_i or b_k is set on one and only one potentiometer; coefficients exceeding unity in absolute value are obtained through combinations of potentiometers and amplifier input gains [e.g., $a_k = 10(a_k/10)$].

1-9. The Impulse Response.[19] While it is clearly impossible to produce impulse voltages in an analog computer, the impulse response $z(t) = h_+(t)$ of a linear system described by

$$z(t) = \frac{1}{p^n + a_{n-1}p^{n-1} + \cdots + a_0}\,x(t) \qquad \left(n > 0;\ p \equiv \frac{d}{dt}\right) \tag{1-15}$$

is bounded and continuous for $t > 0$. Moreover, Laplace transformation of

$$\left.\begin{array}{r} (p^n + a_{n-1}p^{n-1} + \cdots + a_0)h_+(t) = \delta_+(t) \\ \text{with } h_+, ph_+, p^2h_+, \ldots, p^{n-1}h_+ = 0 \text{ for } t = 0 \end{array}\right\} \qquad \left(n > 0;\ p \equiv \frac{d}{dt}\right) \tag{1-16}$$

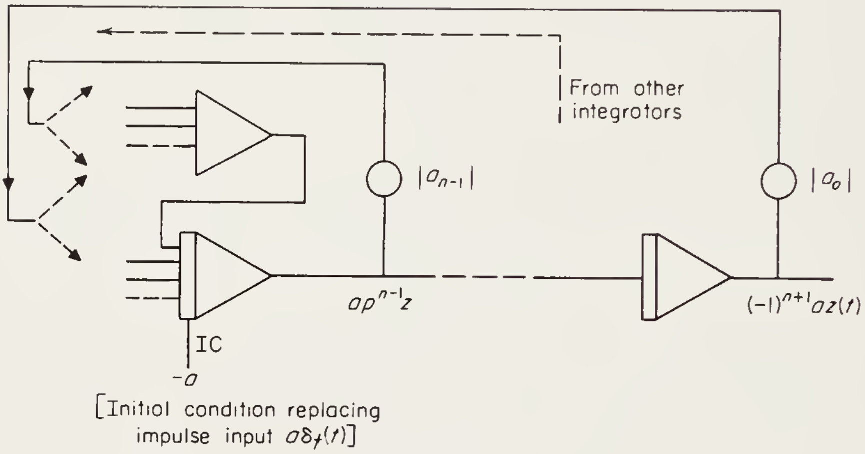


Fig. 1-6. Computer setup producing the impulse response

$$(-1)^{n+1}az(t) = \frac{(-1)^{n+1}a}{p^n + a_{n-1}p^{n-1} + \cdots + a_0}\,\delta_+(t)$$

(*From G. A. Korn, Random-process Simulation and Measurements, McGraw-Hill, New York,* 1966.)

shows that $h_+(t)$ is identical with the solution of the problem

$$\left.\begin{array}{r} (p^n + a_{n-1}p^{n-1} + \cdots + a_0)h_+(t) = 0 \\ \text{with } h_+, ph_+, p^2h_+, \ldots, p^{n-2}h_+ = 0 \text{ and } p^{n-1}h_+ = 1 \text{ for } t = 0 + 0 \end{array}\right\}$$

$$\left(n > 0;\ p \equiv \frac{d}{dt}\right) \tag{1-17}$$

This is easily implemented by the computer setup of Fig. 1-6. *We see that the impulse input is simply replaced by an initial-condition setting on a single integrator.*

The same technique applies also to operations of the more general form (11), provided that $n > m$, so that $H(\infty)$ and $h_+(t)$ are bounded.

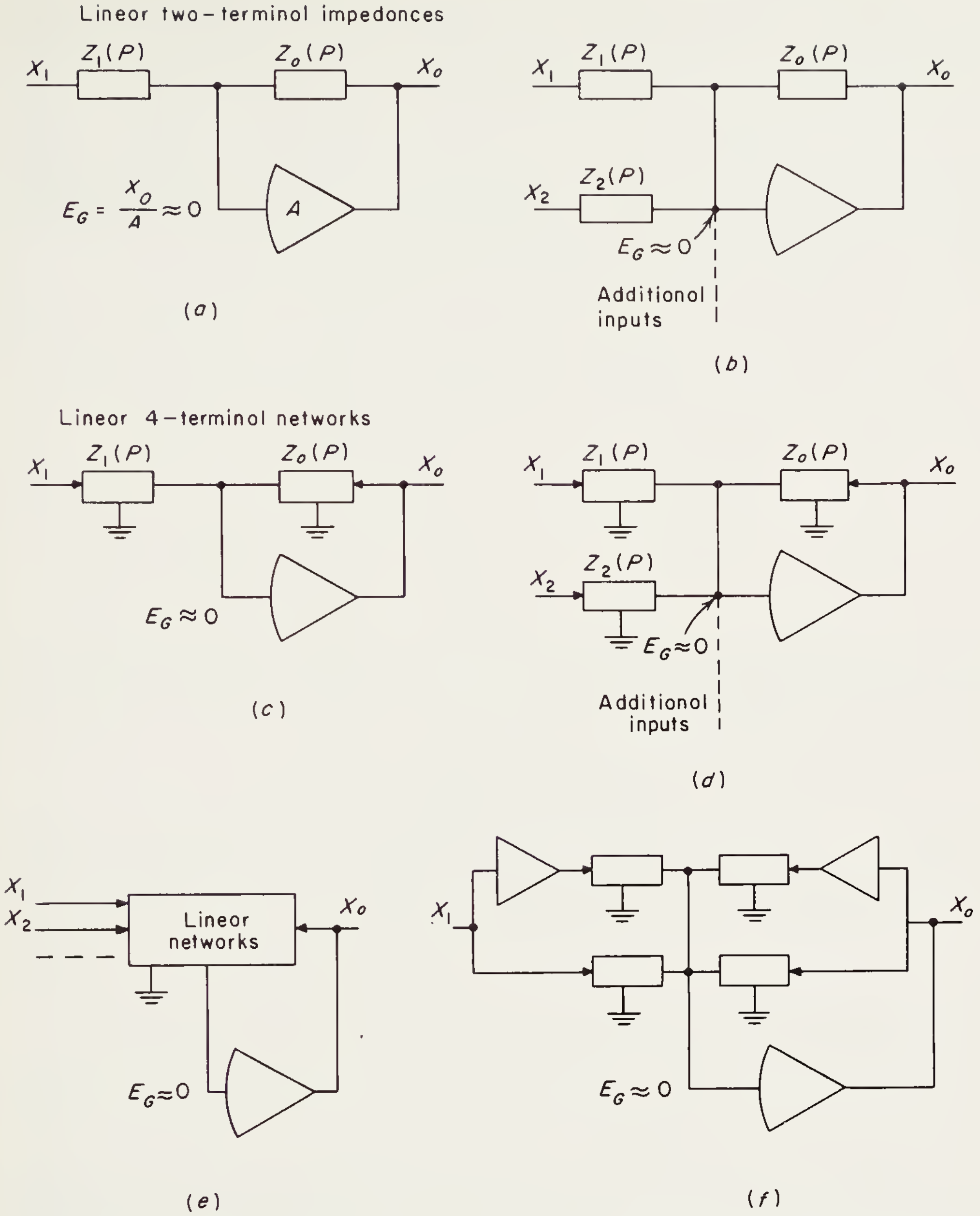


Fig. 1-7. Parallel-feedback operational amplifiers. Each circuit is assumed to be stable; assuming sufficient feedback loop again, each summing-point voltage E_G will be small throughout the working-frequency range.

In this case, we simply generate the impulse response $z(t)$ as in Fig. 1-6 and then produce

$$y(t) = (b_m p^m + b_{m-1} p^{m-1} + \cdots + b_0) z(t) \qquad (m < n)$$

in the manner of Fig. 1-5*b*.

1-10. Operational-amplifier Method. Simple operations of the form (11) are readily implemented by the well-known operational-amplifier circuits of Fig. 1-7. We note that each *impedance* $Z_k(j\omega)$ in Fig. 1-7*a* and b relates a voltage $X_k(\tau)$ and a current $i_k(\tau)$ by a differential equation

$$i_k(\tau) = \frac{1}{Z_k(P)} X_k(\tau) \qquad \left(P \equiv \frac{d}{d\tau}\, ; k = 0,1,2, \ldots\right) \tag{1-18}$$

In each operational amplifier, high-gain degenerative feedback forces the *summing-point voltage* E_G close to ground potential ($E_G \approx 0$). Hence the summing-point node equation in Fig. 1-7*a*,

$$\frac{1}{Z_0(P)} X_0 + \frac{1}{Z_1(P)} X_1 = 0 \tag{1-19}$$

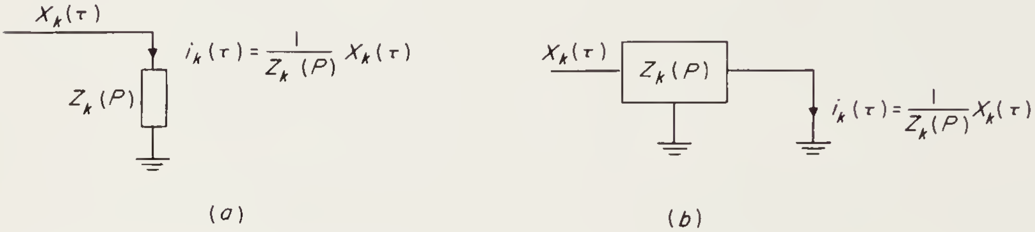

$X_k(\tau)$
$Z_k(P)$
$i_k(\tau) = \frac{1}{Z_k(P)} X_k(\tau)$
(a)
$X_k(\tau)$
$Z_k(P)$
$i_k(\tau) = \frac{1}{Z_k(P)} X_k(\tau)$
(b)

Fig. 1-8. A two-terminal impedance (*a*) and the short-circuit transfer impedance of a network (*b*) as linear operators.

implies the important performance equation

$$X_0(\tau) = -\frac{Z_0(P)}{Z_1(P)} X_1(\tau) \tag{1-20}$$

In Fig. 1-7*b*, the summing-point node equation

$$\frac{1}{Z_0(P)} X_0 + \frac{1}{Z_1(P)} X_1 + \frac{1}{Z_2(P)} X_2 + \cdots = 0 \tag{1-21}$$

similarly yields the basic performance equation

$$X_0(\tau) = -Z_0(P) \left[\frac{1}{Z_1(P)} X_1(\tau) + \frac{1}{Z_2(P)} X_2(\tau) + \cdots\right] \tag{1-22}$$

The more general operational-amplifier circuits of Fig. 1-7*c* and *d* employ passive *four-terminal networks* in their input and feedback lines. The *short-circuit transfer impedance* $Z_k(j\omega)$ or $Z_k(P)$ of such a network is again defined by the differential equation (18), where X_k is the input voltage and i_k is the current drawn from the output terminal short-circuited to ground or to a summing point (Fig. 1-8). Hence the idealized performance equations (20) and (22) apply to the operational amplifiers of Fig. 1-7*c* and *d* as well.

The assumption $E_G = 0$ implies infinite operational-amplifier loop gain $|A(j\omega)\beta(j\omega)|$. The effect of finite loop gain and other error sources will be discussed in detail in Chap. 3; Eqs. (20) and (22) will be shown to hold within $100/|A(j\omega)\beta(j\omega)|$ percent for sinusoidal input of circular frequency ω. For typical operational amplifiers, $|A(j\omega)|$ is between 10^4 and 10^8 at zero frequency and reduces to unity between 0.1 and 50 MHz.

Figure 1-7*e* and *f* shows still more general operational-amplifier circuits, and Figs. 1-9 to 1-12 present examples. In particular, Fig. 1-9*a* illustrates the use of *tables of short-circuit transfer impedances* (Table A-1).

Figure 1-13 shows additional operational-amplifier circuits and design formulas. Such circuits have been successfully applied to the design of *active filters*, which do not require inductances and are therefore especially suitable for integrated-circuit implementation. Note, however, that the same integrated-circuit technology also permits very inexpensive fabrication of the multiamplifier circuits needed for differential-analyzer-type filter implementation (Sec. 1-8). Differential-analyzer-type filter networks employ only simple phase inverters, summers, and integrators, which will, in general, have higher feedback ratios than operational amplifiers with complex feedback networks; this makes the filter performance less dependent on the gains of the inexpensive amplifiers used (Sec. 3-16).[8] Altogether, calculation of filter-parameter *sensitivity* to changes in active and passive circuit components is vital for the design of practical active *RC* filters; this subject is discussed in detail in Ref. 5.

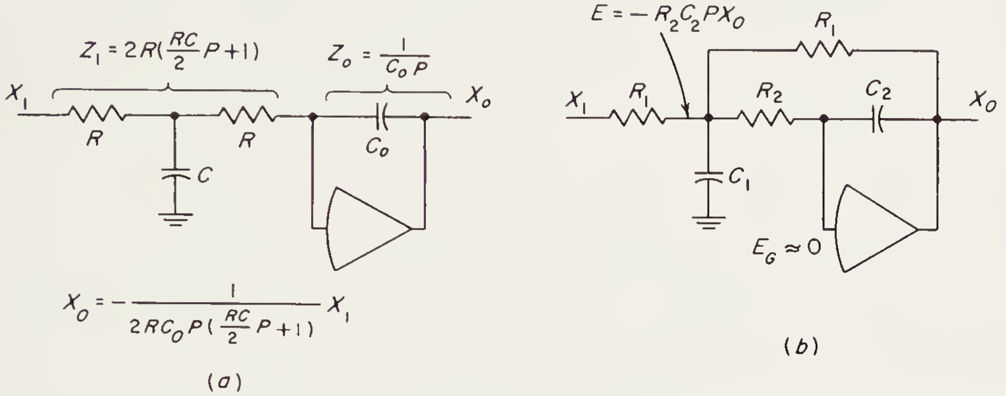


Fig. 1-9. Examples showing operational-amplifier design with the aid of transfer-impedance tables (Table A-1) (*a*), and using node equations (*b*), assuming infinite loop gain. In Fig. 1-9*b*,

$$X_0 = -\frac{1}{(1/\omega_n^2)P^2 + (2\xi/\omega_n)P + 1} X_1$$

which describes damped mass-spring such as accelerometers, rate gyros, valves, and many spring- or servo-restrained transducers and measuring devices. Note that the integrator input voltage must be $-RCPY$ if high loop gain is assumed; one requires, then, only one node equation, that for the integrator-input node, to find

$$\omega_n = \sqrt{\frac{1}{R_1R_2C_1C_2}} \qquad \xi = \left(\frac{R_1}{2} + R_2\right)\sqrt{\frac{C_2}{C_1}\frac{1}{R_1R_2}}$$

The price paid for circuit simplicity is the interaction of adjustments of the natural circular frequency ω_n and the damping ratio ξ; but design is simplified by nomographs of the type given in the Appendix (Fig. A-1).

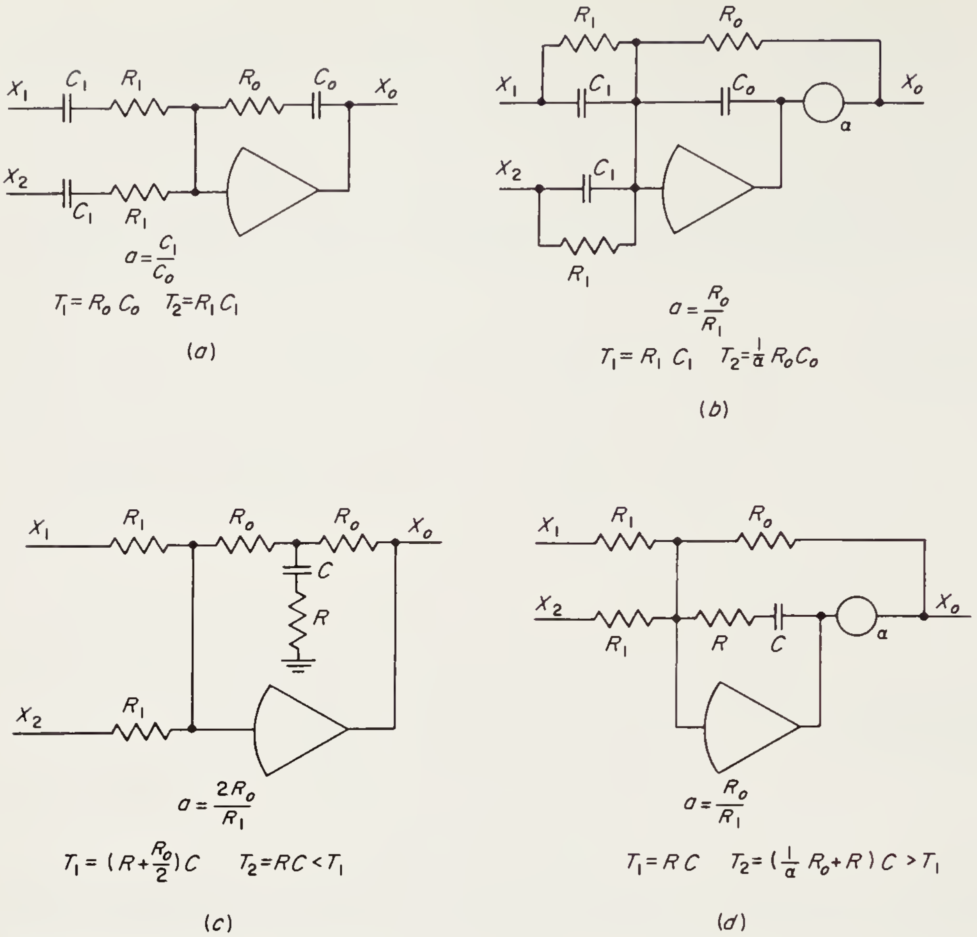


Fig. 1-10. Practical considerations in the design of operational amplifiers. Each circuit implements the relation

$$X_0 = -a \frac{T_1 P + 1}{T_2 P + 1} (X_1 + X_2)$$

e.g., for servomechanism equalization. Figure 1-10*a* is not practical in many applications: there is no d-c feedback, so that stray input voltages can cause amplifier overloads. Figure 1-10*b* permits one to adjust T_2 independently cf a; α must be set with the load connected. The circuits of Fig. 1-10*c*, *d* use simple summing resistors and do not load the sources with capacitance; individual adjustments of the input gains for X_1 and X_2 could be easily added. Many similar circuits can be designed with the aid of Table A-1.

SOME PROGRAMMING TRICKS

1-11. How to Avoid Division. Since feedback division circuits (Fig. 1-2 and Sec. 6-16) often cause stability and scaling problems, it may be useful to transform a given problem so as to avoid division. Thus, the differential equation

$$\frac{dy}{dx} = \frac{f(x,y)}{g(x,y)}$$

is equivalent to the system

$$\frac{dx}{dt} = ag(x,y) \qquad \frac{dy}{dt} = af(x,y)$$

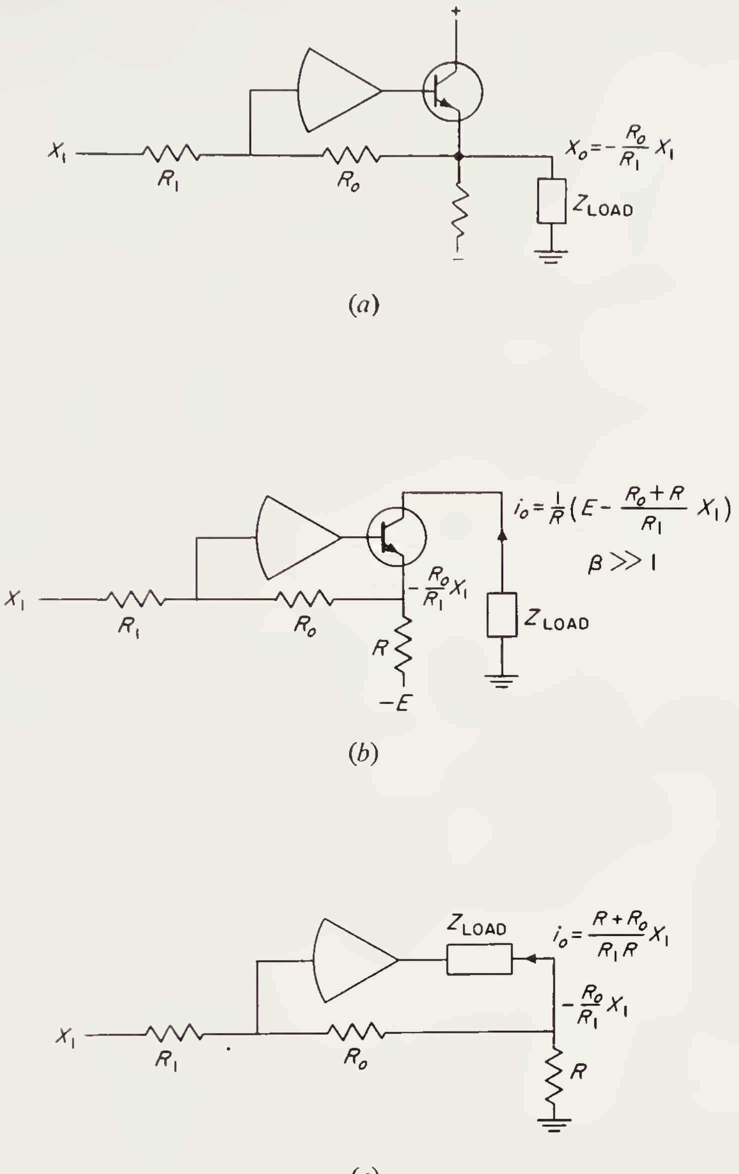


Fig. 1-11*a* to *c*. Voltage regulator (*a*) and current regulators (*b*), (*c*) using operational amplifiers. In each case, either vacuum-tube or transistor output stages can be used; a suitable power output stage may be added to an existing d-c amplifier. In Fig. 1-11*b*, E must be a stable reference voltage; ordinary transistors or vacuum-tube cathode followers yield current in one direction only (reverse the signs of E and i_0 if a *pnp* transistor is used). Substitution of a symmetrical transistor could permit bidirectional operation. The current generators (*b*), (*c*) are useful for driving special networks, electronic switches (Sec. 5-5), and electromagnets requiring accurate control of magnetomotive force, e.g., oscillographs and cathode-ray-tube deflection yokes. Relatively large currents (25 amp) have been regulated by such circuits. Note that the power-transistor output stages required for such large currents must themselves be driven by power amplifiers; so that a cascade of emitter followers may be required.

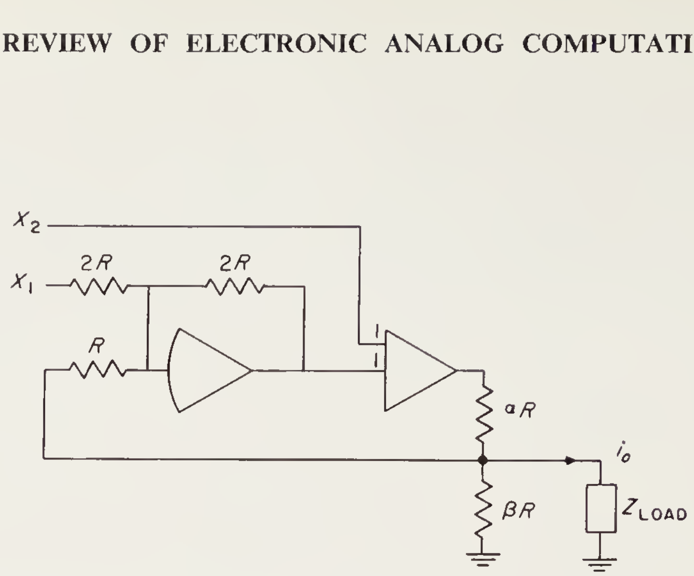


Fig. 1-11*d*. Two-amplifier current generator. The output current i_0 and the source impedance Z_S are given by

$$i_0 = \frac{X_1 - X_2}{\alpha R}\,\frac{1}{1 + Z_L/Z_S} \qquad Z_S = \frac{\alpha\beta R}{\alpha\beta + \alpha - \beta}$$

if d-c amplifier gain is practically infinite. The source impedance becomes infinite if $\alpha\beta + \alpha - \beta = 0$, or if $\beta = \infty$, $\alpha = 1$.

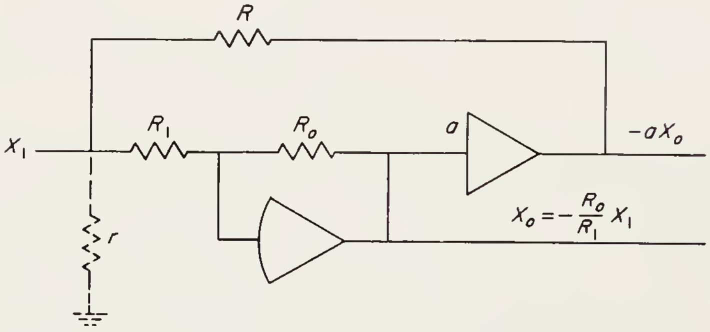


Fig. 1-11*e*. In this *bootstrap unloading circuit*, the "bootstrap current" through R accurately balances the current drawn from the input-voltage source by r and R_1 if

$$R = \frac{r}{r + R_1}(aR_0 - R_1)$$

so that the source sees an infinite impedance. r may be a load external to the bootstrap circuit; if $r = \infty$, one must have $R = aR_0 - R_1$ for infinite-impedance input. (If R is decreased beyond the finite-impedance condition in Fig. 1-11*e*, the source sees a *negative resistance;* in practice, circuit stability will now depend on the source impedance. Substitution of reactive network components for r, R_1, R_0, and R yields unloading circuits for reactive loads, and circuits simulating negative reactances; see also Ref. 5.) Practically attainable input impedances exceed 1,000M; the low-impedance outputs X_0 and $-aX_0$ can drive measuring or computing devices.

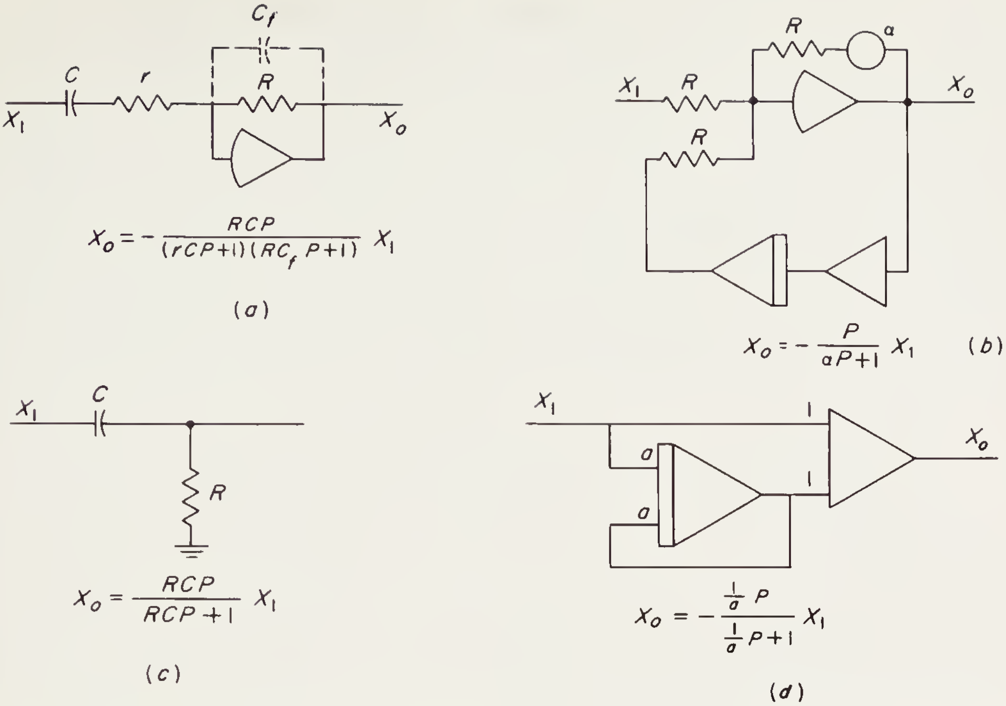

C_f
C
r
R
X_1
X_o
$X_o = -\frac{RCP}{(rCP+1)(RC_f P+1)} X_1$
(a)
R
a
X_1
R
X_o
R
$X_o = -\frac{P}{aP+1} X_1$
(b)
X_1
C
R
$X_o = \frac{RCP}{RCP+1} X_1$
(c)
X_1
1
X_o
a
1
a
$X_o = -\frac{\frac{1}{a}P}{\frac{1}{a}P+1} X_1$
(d)

Fig. 1-12a to d. Practical differentiating circuits. In practice, circuit (*d*) may be less noisy than (*b*) and (*c*).

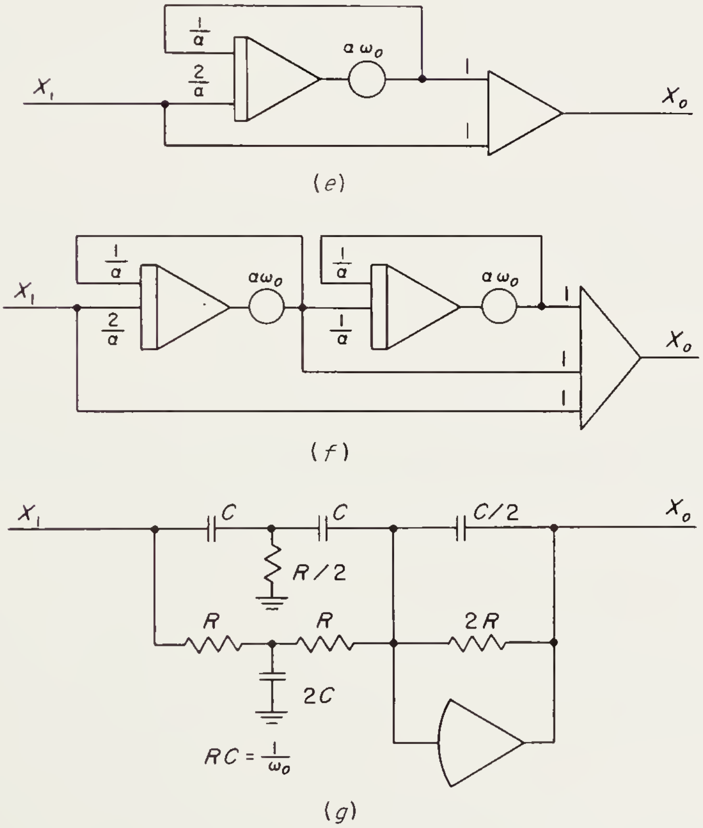


Fig. 1-12e to g. Operational-amplifier phase shifter and notch filters. The phase shifter (*e*) implements $X_0 = \frac{\omega_0 - P}{\omega_0 + P} X_1$; for steady-state sinusoidal input, X_0 lags X_1 by 2 arctan (ω/ω_0). The notch filters (*f*) and (*g*) implement $X_o = -\frac{P^2 + \omega_0^2}{(P + \omega_0)^2} X_1$.

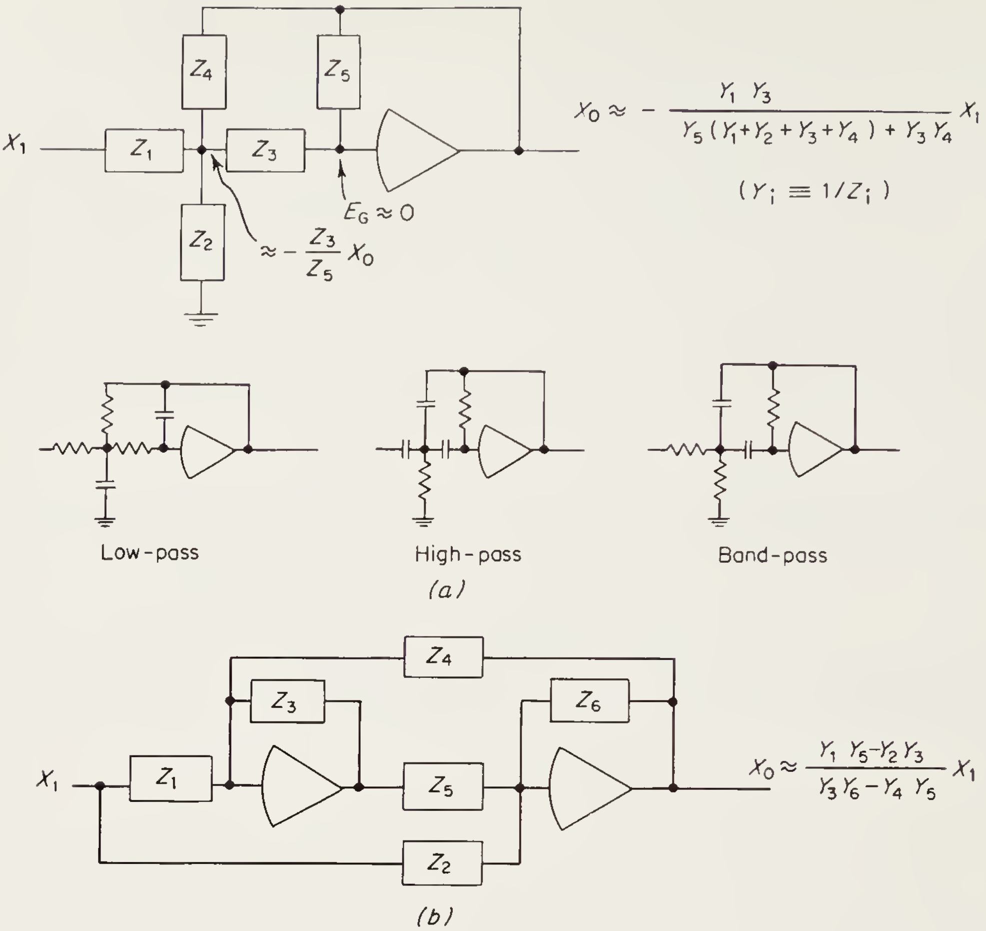

X_1
Z_1
Z_2
Z_3
Z_4
Z_5
$E_G \approx 0$
$\approx -\frac{Z_3}{Z_5} X_0$
$X_0 \approx -\frac{Y_1 Y_3}{Y_5(Y_1+Y_2+Y_3+Y_4)+Y_3 Y_4} X_1$
$(Y_i \equiv 1/Z_i)$
Low-pass
High-pass
Band-pass
(a)
X_1
Z_1
Z_2
Z_3
Z_4
Z_5
Z_6
$X_0 \approx \frac{Y_1 Y_5 - Y_2 Y_3}{Y_3 Y_6 - Y_4 Y_5} X_1$
(b)

Fig. 1-13. Miscellaneous operational-amplifier circuits suitable for active *RC* filters; note that Fig. 1-9*b* is a special case of Fig. 1-13*a*. Admittances Y_i corresponding to each two-terminal impedance Z_i are given by $Y_i = 1/Z_i$; relations are exact for infinite loop gains (based on Ref. 5).

where t is a new independent variable, and a is a constant; we have $y = y(0)$ and $x = 0$ for $t = 0$. Problems in which $f = g = 0$ for some value of t may or may not lead to difficulties.

More generally, systems expressible in the form

$$\frac{dy_i}{dx} = \frac{f_i(x,y_1,y_2, \ldots ,y_n)}{g_i(x,y_1,y_2, \ldots ,y_n)} \qquad (i = 1,2, \ldots ,n) \tag{1-23}$$

are transformed into

$$\left.\begin{aligned} \frac{dx}{dt} &= a \prod_{k=1}^{n} g_k(x,y_1,y_2, \ldots ,y_n) \\ \frac{dy_i}{dt} &= af_i(x,y_1,y_2, \ldots ,y_n) \prod_{k \neq i}^{n} g_k(x,y_1,y_2, \ldots ,y_n) \end{aligned}\right\}$$

with $x = 0$ for $t = 0$. Note also that many differential equations

$$\frac{dy_i}{dx} = F_i(x, y_1, y_2, \ldots, y_n)$$

where F_i is an unbounded function, can be rewritten in the form (23).

1-12. Proper Use of Function Generators. A general function $y = f(x)$ will be represented in the computer by the scaled function-generator output

$$Y = a_y y = a_y f\left[\frac{1}{a_x}(a_x x)\right] = F[(a_x x)] = F(X)$$

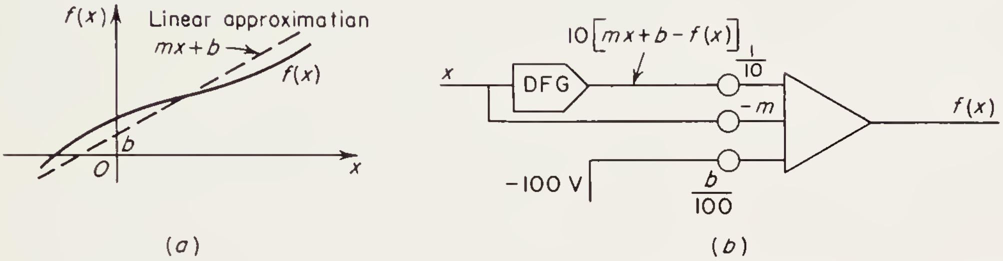


Fig. 1-14. A typical function $f(x)$ (*a*), and generation of $f(x)$ as the sum of a linear approximation $mx + b$ and a small correction function (*b*). Note that all function-generator errors are divided by 10. (*From H. D. Huskey and G. A. Korn, Computer Handbook, McGraw-Hill, New York*, 1962.)

corresponding to the function-generator input $a_x x = X$. If, for example, $|x| < 5$, $y = x^3 + x$ can be represented by

$$(0.5y) = \frac{1}{16{,}000}(20x)^3 + \frac{1}{40}(20x)$$

Arbitrary-function generators, such as diode function generators, tapped potentiometers, or curve followers (Secs. 4-1 and 4-15) are usually less accurate than linear computing elements and multipliers. To minimize the effects of function-generator errors:

1. Attempt to scale function-generator input and output voltages so as to utilize the full voltage range available.
2. Whenever possible, arbitrary-function generators should supply only *corrections* to analytic (most frequently linear) approximations of the desired function.

In Fig. 1-14, a typical function is generated as the sum of a linear approximation $mx + b$ and a correction function; note how the function-generator output, and hence the function-generator error, is divided by a factor of 10. Many diode function generators (Sec. 4-15) already comprise accurate linear network components for generating $mx + b$, so that their diode networks generate only the correction function.

1-13. Generalized Integration. Operational-amplifier integrators integrate the sum of their input voltages with respect to the computer time τ (real time), which serves as the independent machine variable. It is, however, possible to obtain integrals of a voltage $Y = Y(\tau)$ with respect to a dependent machine variable $X = X(\tau)$ in the form

$$Z = \int_0^\tau Y\,dX + Z(0) = \int_0^\tau \left(Y \frac{dX}{d\tau}\right) d\tau + Z(0)$$

if a voltage proportional to the time derivative $dX/d\tau$ of X is available in the computer (*generalized integration*, Fig. 1-15). Substitution of "generalized integrators" like that of Fig. 1-15 in Fig. 1-4*d* yields functions of X (see also Sec. 6-24).

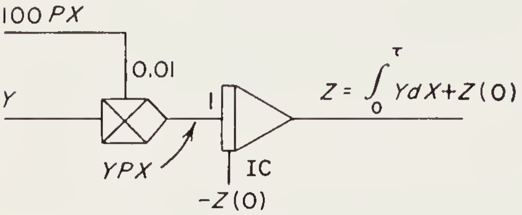


Fig. 1-15. Integration with respect to a machine variable X when a voltage proportional to the time derivative $dX/d\tau = PX$ is available in the computer. Substitution of such "generalized integrators" in Fig. 1-4*d* yields functions of X.

REFERENCES AND BIBLIOGRAPHY

General

1. Rubin, A. I.: Analog-computer Programming, Scaling, and Problem Preparation, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
2. ———: Function Generators, *PCC Rept.* 116, Electronics Associates, Inc., Computing Center, Princeton, N.J., 1959.
3. Fisher, M. E.: Avoiding the Need for Dividing Units, *J. Sci. Instr.*, August 1957.
4. Hausner, A.: Parametric Techniques for Eliminating Division and Treating Singularities in Computer Solutions of Ordinary Differential Equations, *IRETEC*, February 1962.

Active *RC* Filters

5. Huelsman, L. P.: *Theory and Design of Active RC Circuits*, McGraw-Hill, New York, 1968.
6. Geffe, P. R.: Toward High Stability in Active Filters, *IEEE Spectrum*, May 1970.
7. Moschytz, G. S.: Inductorless Filters: A Survey, *IEEE Spectrum*, September 1970.
8. Kerwin, W. J., et al.: State-variable Synthesis for Insensitive Integrated-circuit Transfer Function, *IEEE J. Solid-state Circuits*, September 1967.
9. Lovering, W. P.: Analog-computer Simulation of Transfer Functions, *Proc. IEEE*, March 1965.

Selected Books on Analog/Hybrid Computation

10. Bekey, G. A., and W. J. Karplus: *Hybrid Computation*, Wiley, New York, 1968.
11. Fifer, S.: *Analogue Computation* (4 vols.), McGraw-Hill, New York, 1961.
12. Giloi, W., and R. Lauber: *Analogrechnen*, Springer, Berlin, 1963.
13. Gilbert, C. P.: *The Design and Use of Electronic Analogue Computers*, Chapman and Hall, London, 1964.
14. Huskey, H. D., and G. A. Korn: *Computer Handbook*, McGraw-Hill, New York, 1962.
15. Jackson, A. S.: *Analog Computation*, McGraw-Hill, New York, 1960.
16. Johnson, C. L.: *Analog Computer Techniques*, 2d ed., McGraw-Hill, New York, 1963.
17. Kley, A.: *Elektronische Hybridrechner*, Telekosmos, Stuttgart, 1969.
18. Kogan, B. Y.: *Electronic Analog Computers*, Fismatgis, Moscow, 1963.

19. Korn, G. A.: *Random-process Simulation and Measurements*, McGraw-Hill, New York, 1966.
20. Levine, L.: *Methods for Solving Engineering Problems Using Analog Computers*, McGraw-Hill, New York, 1964.
21. McLeod, J.: *Simulation: The Dynamic Modeling of Ideas and Systems with Computers*, McGraw-Hill, New York, 1968.
22. MacKay, D. M., and M. E. Fisher: *Analogue Computing at Ultra-high Speed*, Wiley, New York, 1962.
23. Rogers, A. E., and T. W. Connolly: *Analog Computation in Engineering Design*, McGraw-Hill, New York, 1960.
24. Tomovič, R., and W. J. Karplus: *High-speed Analog Computers*, Wiley, New York, 1962.
25. Hausner, A.: *Analog and Analog/Hybrid Computer Programming*, Prentice-Hall, Englewood Cliffs, N.J., 1971.

Periodical Literature on Analog Computation

The most important periodical sources are:

1. *Simulation* (Simulation Councils, Inc., La Jolla, Calif.)
2. *IEEE Transactions on Computers* (*IEEETC*; earlier issues identified first as *IRETEC* and later as *IEEETEC*). Contains abstracts and bibliographies.
3. *Automatika i Telemekhanika* (translated by Instrument Society of America; also contains periodical bibliographies on analog computation).
4. *Annales de l'Association Internationale pour le Calcul Analogique* (*AICA*), Presses Académiques Européennes, Brussels; contains bibliographies. See also the *Proceedings of AICA Conferences* published every three years.
5. (British) *Computer Journal.*
6. *Rechentechnik.*

Consult also current issues of *Electronics*, *Electronic Design*, *Electronic Design News* (*EDN*), *Electronics Letters*, *Control Engineering*, *Instruments and Control Systems*, and *IEEE Computer Journal.*

CHAPTER 2

ITERATIVE ANALOG COMPUTATION AND HYBRID ANALOG-DIGITAL COMPUTERS

ITERATIVE ANALOG COMPUTATION

2-1. A Critical Appraisal of Analog Computation. *Component accuracies* of individual analog computing elements vary between 5 and 0.005 percent of half-scale (usually 10 or 100 volts, Sec. 3-1). Costs increase rapidly if component accuracies better than 0.1 percent are desired. Solution accuracies for a *complete problem* depend critically on the problem and on the computer setup as well as on the component accuracies, since errors may cancel or be compounded. The overall accuracy can be practically equal to the component accuracy in problems involving linear differential equations with constant coefficients if the solutions comprise only well-damped modes (linear-servo simulation).

Special-purpose electrical analog computing circuits fit, of course, neatly and directly into electrical control and instrumentation systems. *General-purpose* analog computers serve mainly for simulation (i.e., experimentation with models) of dynamical systems described by ordinary differential equations (e.g., aerospace vehicles, chemical processes). Traditionally, analog simulation has functioned as an on-line, direct aid to the investigator's thinking process. Simple patchcord connections and attenuator settings permit creation and ready modification of a "live mathematical model," which can be operated and changed to study performance and interaction of processes and systems. Model operation can be conveniently speeded up or slowed down (*time scaling*, Sec. 1-6*b*). With the computer running on a 1:1 time scale, actual physical components of a system under investigation can take the place of a corresponding block of computing elements (*partial system testing*).

But the traditional advantage of "hands-on" analog simulation has been overtaken by the development of *all-digital on-line simulation systems.*[1–3] On-line digital simulation systems are *more* convenient to program than analog computers, permit automatic sequencing of computations, automatic report preparation, and, above all, scale-factor-free floating-point operation;[1,2] partial system tests are readily possible.[3]

We conclude, therefore, that *the sole advantage of general-purpose analog computers is their ability to solve fairly large and complicated systems of ordinary differential equations at relatively very high speed.* Component accuracies within 0.1 percent of half-scale are possible at analog-signal frequencies of 10 to 20 KHz (Secs. 3-35, 5-16, 6-2, and 7-2), although 0.5 to 2 KHz is more common. A still more important speed advantage of analog over conventional digital computation is realized especially in larger simulations: *the analog computer employs multiple "parallel" computing elements for each mathematical operation* and can thus simulate a larger system as quickly as a smaller system. Typical simulation problems involve 50 to 300 operational amplifiers; as many as 1,000 amplifiers have been used.

Modern analog/hybrid computers are fast, reliable, and convenient, but not cheap. They will compete successfully with digital simulation where their inherent speed advantage is really important, viz., in the following applications:

1. Simulation problems requiring *thousands* of differential-equation-solving computer runs for parameter optimization, contingency studies, and statistical evaluations
2. *Partial system tests requiring* 1:1 *time-scale simulation of very large systems* (e.g., simulation of a space-vehicle system with real pilots and instruments in the loop)
3. Special techniques for solving *partial differential equations* (see the bibliography for Chap. 8)

In all practical applications, general-purpose analog computation benefits so greatly from the assistance of digital techniques that few "pure" analog computers survive. Essentially all serious applications employ *hybrid analog-digital computation.*

2-2. Evolution of Iterative Analog Computation. (a) How to Benefit from Analog-computer Speed. The possibility of operating many analog computing elements in parallel permits real-time simulation of large dynamical systems. As a matter of fact, computing-element bandwidths make it possible to simulate many systems on a speeded-up ("fast") time scale. Unlike the situation in digital machines, this computing speed is difficult to trade off for increased accuracy; how, then, can we benefit from analog computing speed? Figure 2-1 illustrates some possible answers.

In iterative parameter studies with "slow" analog computers, the operator looks at the solution records of a differential-equation-solving run, adjusts parameters to new values, makes another run, etc. For better utilization of analog-computer bandwidth, *repetitive analog computers* (Fig. 2-1*a*) quite early repeated the solution procedure rapidly (10 to 50, later up to 4,000 solutions per second) by periodically returning the integrators to initial conditions with periodic RESET

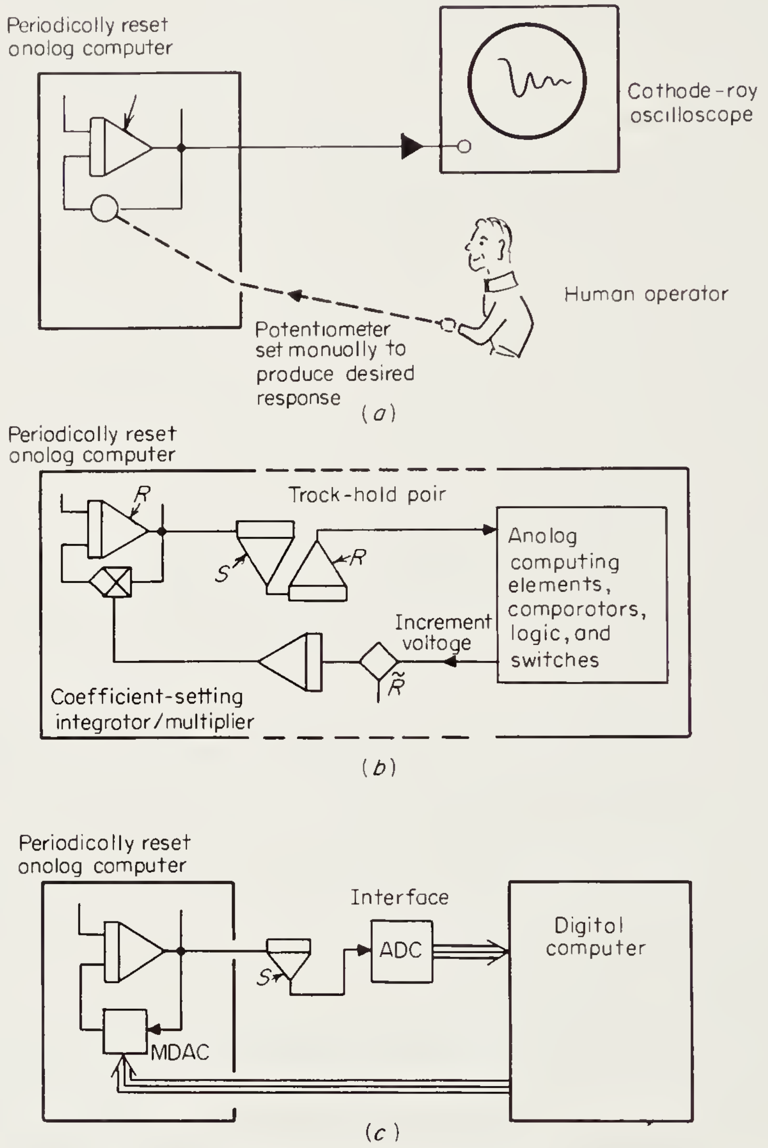

Periodically reset analog computer
Cathode-ray oscilloscope
Human operator
Potentiometer set manually to produce desired response
(a)
Periodically reset analog computer
R
Track-hold pair
S
R
Analog computing elements, comparators, logic, and switches
Increment voltage
$\tilde{R}$
Coefficient-setting integrator/multiplier
(b)
Periodically reset analog computer
Interface
ADC
S
MDAC
Digital computer
(c)

Fig. 2-1. Evolution of iterative analog computation: repetitive analog computer with manual parameter adjustments, (*a*); iterative differential analyzer with track-hold analog memory and patched logic (*b*); and hybrid analog-digital iteration (*c*).

pulses. During each cycle, all machine variables vary in the prescribed manner and are then reset to their initial values. Results can now be presented on an ordinary cathode-ray oscillograph whose sweep frequency equals the computer repetition rate. The rapid operation of such machines permits immediate observation of the effects of parameter changes on solutions.

The primitive repetitive analog computer still wastes analog-computer bandwidth because the machine can generate new solutions faster than the operator can make parameter adjustments. An *iterative differential analyzer* will automatically perform *successive analog-computer runs which can utilize stored results of earlier runs, so that iterative computation becomes possible.* Iterative differential analyzers (and practically all modern analog/hybrid computers belong in this category) add a number of essentially hybrid analog-digital computing elements to the basic analog computer. *Track-hold circuits* timed by digital timing and logic circuits sample analog-computer solutions; *analog comparators* sense sign changes in analog-computer solutions or track-hold outputs; *patched digital logic*, usually brought out to a separate *control or digital patchbay*, can logically combine comparator and timing signals to operate integrator mode-control switches, track-hold circuits, and electronic switches *so as to modify the analog-computer program.* Parameters can be controlled through parameter-setting integrators or, preferably, by digital circuits (Figs. 2-1*b* and 2-2).

With the availability of more and more inexpensive digital computers, *much of the digital logic and analog circuitry closing an iterative differential analyzer loop can be replaced by a digital computer*, which permits much more convenient and elaborate programming of the iteration (Fig. 2-1*c*). We now have a true hybrid analog-digital computer. This will require addition of new hardware, viz., an analog/digital interface consisting of *analog-to-digital converters* (ADCs) and *digital-to-analog converters* or *multiplying digital-to-analog converters* (DACs or MDACs), plus *digital timing and control circuits* for this interface (Secs. 5-6 to 5-15 and 7-8 to 7-12).

(b) Iterative-differential-analyzer Operation (see Secs. 8-13 to 8-16, and 8-24 for examples). Iterative differential analyzers, like digital computers, are programmed through a series of *subroutines*. A subroutine is a sequence of operations, such as an analog-computer run or a number of repetitive analog-computer runs. We associate each subroutine with a digital (binary) *control variable* U_i representing the state of a control relay or flip-flop. The subroutine proceeds when $U_i = 1$; $U_i = 0$ "resets" the computing elements involved in the subroutine (e.g., integrators, counters) for renewed use. Note that the complementary control variable $\tilde{U}_i$ (0 for $U_i = 1$, 1 for $U_i = 0$) may also

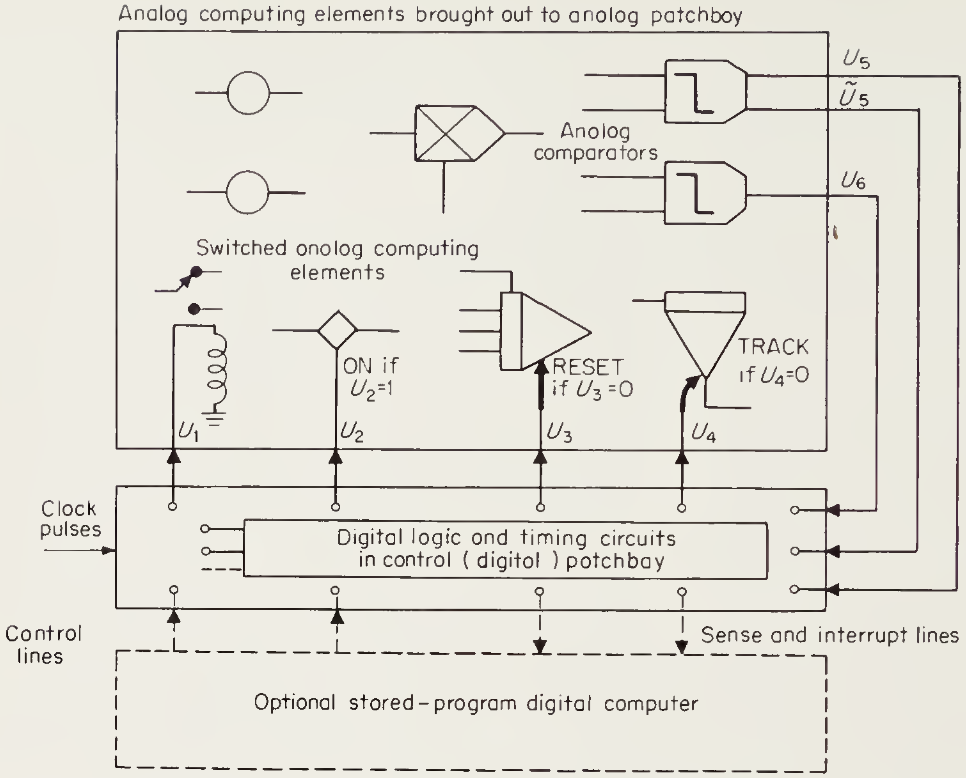

Analog computing elements brought out to analog patchbay
U_5
$\tilde{U}_5$
Analog comparators
U_6
Switched analog computing elements
ON if U_2=1
RESET if U_3=0
TRACK if U_4=0
U_1
U_2
U_3
U_4
Clock pulses
Digital logic and timing circuits in control (digital) patchbay
Control lines
Sense and interrupt lines
Optional stored-program digital computer

Fig. 2-2. Iterative-differential-analyzer system. Operations are mainly analog-computer runs "ordered" by electronic switches or relays controlled by binary control variables U_i. Control-variable states are preset by timers and counters and/or "computed" by analog comparators and digital logic. Switches and comparators form digital-analog and analog-digital interfaces.

define a subroutine. Subroutines can be nested; i.e., they may involve component subroutines.

EXAMPLES: An ordinary differential-analyzer run and the subsequent RESET period are subroutines controlled by the states $\tilde{R} = 0$, $\tilde{R} = 1$ of the COMPUTE relay or switch in a "slow" analog computer and by a RESET flip-flop in a fast repetitive computer. A preset sequence of, say, 1,000 repetitive-computer runs is a subroutine controlled by the output state of a preset run-counter coincidence circuit.

Typical *analog-subroutine changes* are combinations of the following operations:

1. *Switching a group of integrators* from RESET to COMPUTE, or from RESET (TRACK) to HOLD, and vice versa (complementary subroutines)
2. *Switching to new values of parameter or initial-value settings* (e.g., parameter optimization, automatic scale-factor changes)
3. *Switching interconnections* to produce computer-setup changes

Subroutines start and terminate when the corresponding binary control variables change state as logical functions of:

1. External control (manual switching, commands from associated equipment)
2. States of preset timers and run counters
3. Analog-comparator decisions

Appropriate Boolean functions and sequences of these inputs will be implemented by patched electronic logic (Sec. 7-4).

Figure 2-2*a* illustrates the special *hybrid analog-digital structure* of an iterative differential analyzer. Relays or electronic switches implement analog-subroutine changes under control of digital (binary) control variables U_i and constitute the digital-to-analog interface of our hybrid computer. Analog solutions, in turn, can modify digital control variables, and hence the computer program, by way of analog comparators actuating digital logic. Finally, track-hold circuits (or other analog-memory elements, Secs. 5-6 and 7-4b) make computed analog data available for use in later subroutines, while digital variables can be stored as flip-flop states.

Figure 2-2 also introduces our block-diagram notation for hybrid analog-digital computer setups. Digital (binary) variables can take the *symbolic* values 0 and 1 respectively represented by deenergized and energized relays or digital-module outputs (typically 0 and +4 volts). Binary variables can be stored, complemented, and logically combined by digital circuits. Note, in particular, that:

1. Each SPST relay or electronic switch has a binary control input equal to 1 if and only if the relay or switch is *closed.* SPDT relays have an arrow to indicate the closed state.
2. Each integrator or track-hold circuit has a binary control input (RESET input) equal to 0 if and only if the circuit is in RESET or TRACK.* Reset pulses or levels controlling groups of integrators will usually be labeled $R, R', R'', \ldots$; R is the only control variable in simple "slow" or repetitive analog computation.
3. Three-state (COMPUTE/HOLD/RESET) integrators, if used, have a HOLD control input V as well as a RESET input U.

Comparator decisions can be made in the course of analog-computer runs and may, for instance, terminate a run, command sampling, etc. If there is any choice, however, both comparator decisions and subroutine changes should be made in RESET or HOLD states for improved accuracy.

---

* Our convention is the opposite of that adopted in the journal *Simulation*. Our reason is that $R = 0$ in RESET ensures that integrators cannot be in COMPUTE (and thus cannot drift into overload) when no logic connection is patched into the RESET terminal.

The operation of iterative differential analyzers, like that of digital computers, is conveniently represented by *flow charts.* The flow chart (Fig. 2-3) indicates a sequence of switch-controlled subroutines (rectangular *operation boxes*) and program-stepping decisions (oval and diamond-shaped *decision boxes*). We shall employ oval decision boxes for *preset* decisions determined beforehand by digital timers and/or run counters, and diamond-shaped decision boxes for "computed" decisions involving analog comparators or comparator-actuated digital logic. The flow-chart structure is implemented by the interplay of analog and digital components in the machine. Flow-chart *loops* will represent cycles of repetitive or iterative computations.

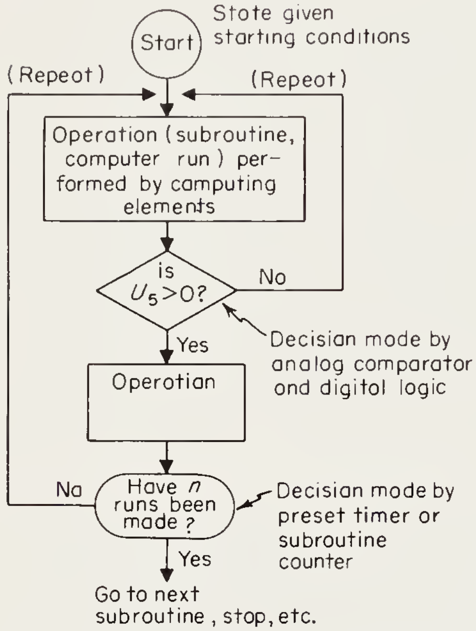

Start
State given starting conditions
(Repeat)
(Repeat)
Operation (subroutine, computer run) performed by computing elements
is $U_5>0$?
No
Yes
Decision made by analog comparator and digital logic
Operation
No
Have n runs been made ?
Yes
Decision made by preset timer or subroutine counter
Go to next subroutine, stop, etc.

Fig. 2-3. Iterative-differential-analyzer flow chart. Rectangular boxes specify operations. Oval decision boxes refer to preset digital-timer and/or counter decisions, while diamond-shaped decision boxes involve analog comparators and/or comparator-actuated digital logic. (*From H. R. Eckes and G. A. Korn, Digital Program Control for Iterative Differential Analyzers, Simulation, February* 1964.)

(c) Analog Memory: Track-hold Pairs. To store the computed value $X(\tau_1)$ of an analog-computer voltage $X(\tau)$ (*point storage*), we track $X(\tau)$ or $-X(\tau)$ with a track-hold circuit and switch into HOLD at the computer time $\tau = \tau_1 = \alpha_t t_1$. If the time derivative $PX = dX/d\tau$ is available in our computer setup, we can also store $X(\tau_1)$ by switching an integrator with input $-PX$ into HOLD. The stored voltage $X(\tau_1)$ may be used:

1. *As a (constant) input or parameter value during the same computer run.* The storage circuit then becomes again available for tracking at the start of the next RESET period.
2. *To set an initial condition for the following computer run.* The storage circuit will be available for renewed tracking at the start of the next COMPUTE period.
3. *As a (constant) input or parameter value during the following computer run.* In this case, the storage circuit will not be free to track again until the end of the next COMPUTE period.
4. During a later RESET or COMPUTE period.

Since a single storage capacitor cannot track and hold simultaneously, we shall require at least *two* storage circuits to transfer information between successive COMPUTE periods. Figure 2-4 shows a *memory pair* comprising two cascaded phase-inverting track-hold circuits. The control pulses S' for track-hold 2 are delayed so that amplifier 2 tracks the hold output of amplifier 1 and stores or "presents" it while amplifier 1 is free to track again. The memory-pair output is a delayed sequence of clean sampled-data steps. Suitably timed pulses can be obtained from a one-shot multivibrator or from a digital clock (Sec. 7-3).

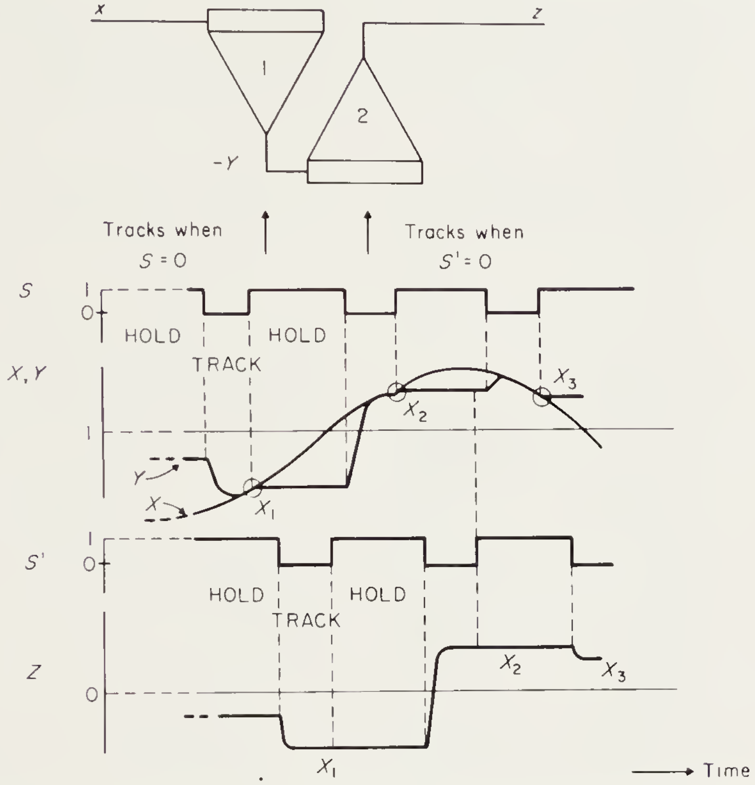


Fig. 2-4. Memory-pair operation. The delayed "resampling" pulses S' are often replaced with complementary pulses $\tilde{S}$. This is permissible because most track-hold circuits switch more quickly into HOLD than into TRACK (Sec. 5-17). No initial-reset circuits are shown.

Figure 2-5*a* illustrates *memory-pair operation for information transfer between successive analog-computer runs.* The TRACK pulses $S' = R$ for track-hold 2 are delayed so that amplifier 2 tracks the HOLD output of amplifier 1 during the computer RESET period and then holds or "presents" the stored voltage during the entire subsequent COMPUTE period. Amplifier 1, in the meantime, is free to track again. Unfortunately, this scheme breaks down whenever the sampling time τ_1 is shorter than the period T_S required for tracking, for now amplifier 1 is already in TRACK during the RESET period (Fig. 2-5*b*). There are two ways out:

1. We can interpose a third track-hold circuit between track-holds 1 and 2 to resample the voltage Y at a more convenient time $\tau_2 > T_S$ (Fig. 2-5*b*).

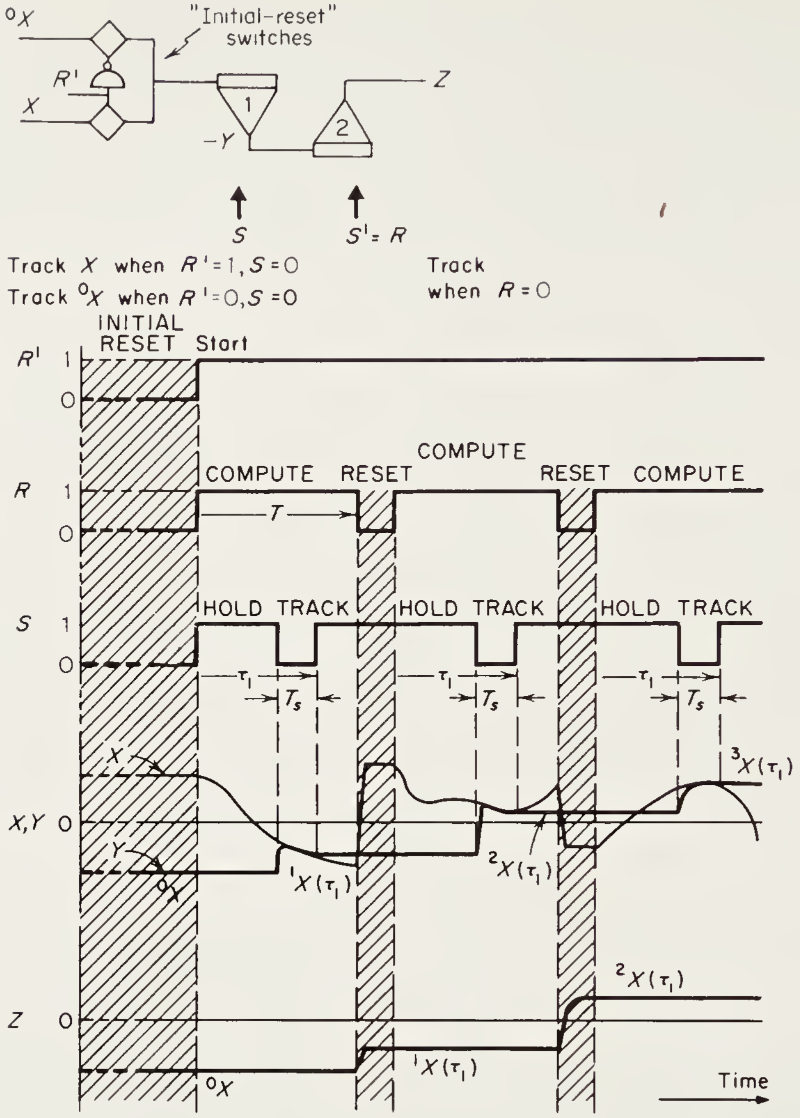
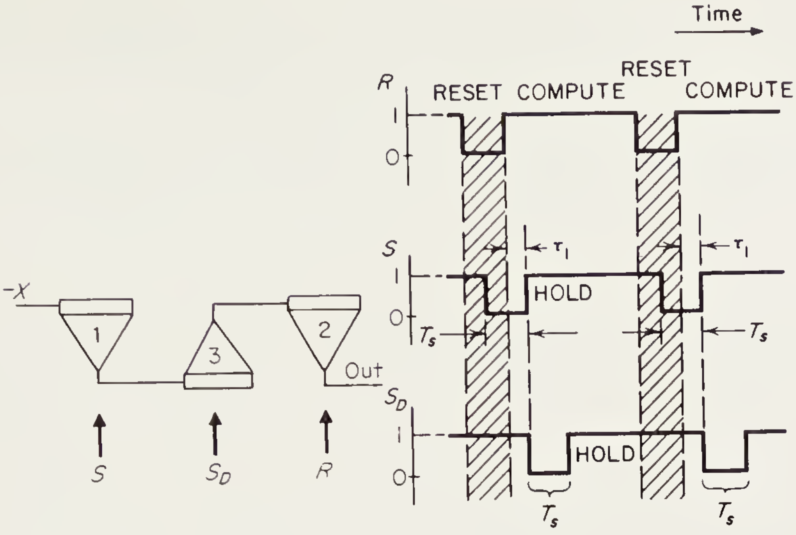


Fig. 2-5*a*. A simple memory pair can present a solution sample $kX(r_1)$ during the entire following iterative-differential-analyzer run if $\tau_1 > T_S$. Note the initial-reset-circuit operation; integrator 1 tracks the initial-reset input when $S = 0$, $R' = 0$.

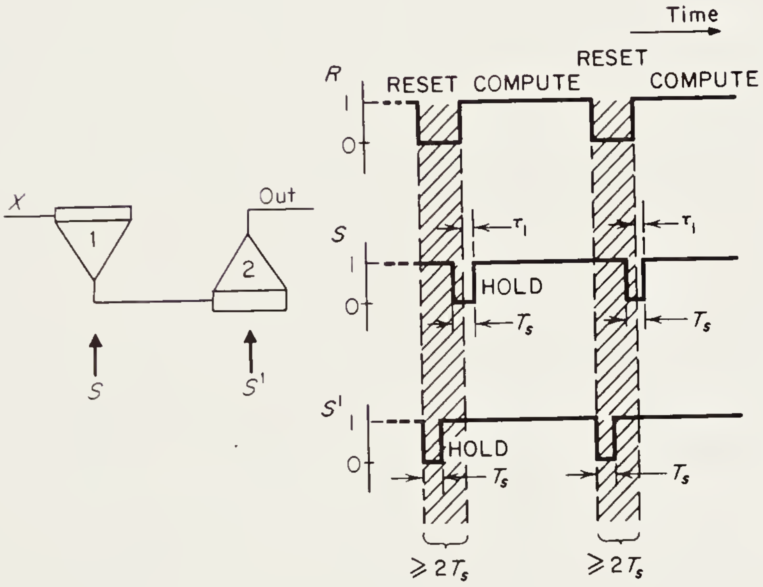
2. We can use a longer computer RESET period (at least equal to $2T_S$) and switch amplifier 2 into HOLD T_S sec after the start of the RESET period ("three-period control," Fig. 2-5*c*).

Both techniques complicate our control circuits (Sec. 7-3). Three-period control permits flexible operation of three-state integrators but tends to waste possibly valuable computing time.

Frequently, a stored voltage $X(\tau_1)$ is required only during the subsequent RESET period, e.g., for setting initial values, or for performing an intermediate subroutine. In this case, a single track-hold circuit suffices for storage if $\tau_1 > T_S$ (Fig. 2-5*a*).



(*b*)



(*c*)

Fig. 2-5*b*, *c*. For $\tau_1 < T_S$, the simple memory scheme of Fig. 2-5*a* breaks down, since the S and R pulses overlap. To present $X(\tau_1)$ during the following computer run, we can either use an extra track-hold circuit with delayed sampling (*b*) or employ three-period control (*c*). No initial-reset circuits are shown.

(d) The INITIAL RESET Mode. In many applications (for examples see Sec. 2-5), the memory-pair output Z in Fig. 2-5*a* must assume a specified initial value ${}^0X$ during the first COMPUTE period. This is achieved by the *initial-reset circuit* shown in Fig. 2-5*a*. The INITIAL RESET mode established by $R' = 0$ implies $R = 0$, $S = 0$; $SW1$ closes, and the memory output Z assumes the correct initial value $Z = {}^0X$. To start the computation, we switch to $R' = R = 1$ and let R and S cycle normally.

TRACK-HOLD OPERATIONS WITH SAMPLED DATA AND SIMULATION OF DIGITAL COMPUTERS

2-3. Sampled-data Sequences and Solution of Difference Equations (see also Sec. 2-4). Let τ represent real time, and let

$$X(\tau) = X(\tau_i) = X_i \qquad (\tau_i < \tau < \tau_{i+1};\ i = 0,1,2, \ldots) \tag{2-1}$$

be the stepwise output of a track-hold circuit, memory pair, or digital-to-analog converter. The step-function voltage $X(\tau)$ represents a *sampled-data sequence* $X_0, X_1, X_2, \ldots$; X_i may be regarded as a function of its integral-valued subscript $i = 0, 1, 2, \ldots$. *Recursion relations* (*difference equations*) relating such data sequences are often encountered in iterative analog computation (Secs. 2-2 and 8-14) and especially in control-system studies requiring *simulation of digital controllers.* In either case, the *best* approach, usually, is to represent data sequences and recursion relations *digitally* on a hybrid computer. Sometimes however, it is expedient to have a method for implementing sampled-data recursion relations on an analog computer (especially if no digital computer is available, or if the available digital computer is too small and/or too slow). Unlike actual digital data, data sequences represented as analog step-function voltages can be neatly added, multiplied, etc., with analog computing elements; and important *difference operations* can be implemented with the aid of *memory pairs* (Sec. 2-4).

Reference to Fig. 2-6 shows that *the stepwise output* $Y(\tau)$ *of a suitably timed memory pair with input* (1) *will represent the input sampled-data sequence delayed by one step, i.e.,*

$$Y_i = Y(\tau_i) = X(\tau_{i-1}) = X_{i-1} = \mathbf{E}^{-1}X_i \qquad (i = 0,1,2, \ldots ;\ X_{-1} = 0) \tag{2-2}$$

The track-hold pair of Fig. 2-6*a* implements the linear *discrete-delay operator* $\mathbf{E}^{-1}$ defined by Eq. (2)* exactly like an analog integrator implements the linear reciprocal-derivative operator $-P^{-1} \equiv -(d/d\tau)^{-1}$. A cascade of memory pairs synchronized in the manner of Fig. 2-6*b* will, then, produce successively delayed data sequences

$$\left.\begin{aligned} \mathbf{E}^{-1}X_i &= X_{i-1} \\ \mathbf{E}^{-1}X_{i-1} &= E^{-2}X_i = X_{i-2} \\ \cdots&\cdots\cdots\cdots \end{aligned}\right\} \qquad (i = 0,1,2, \ldots ;\ X_i = 0 \text{ for } k < 0) \tag{2-3}$$

---

* In the Laplace transform domain, the operator P^{-1} corresponds to multiplication by s^{-1}. Similarly, if we introduce z transforms $\mathscr{Z}[X(s)] = \sum_{k=0}^{\infty} X_k z^{-k}$ of our data sequences X_i, then the operator $\mathbf{E}^{-1}$ corresponds to multiplication by z^{-1}.

and the input data sequence X_0, X_1, X_2, . . . will proceed, step by step, through the memory chain (analog shift register, "bucket-brigade" time delay).

We next employ a track-hold pair to solve a *first-order difference equation* or recurrence relation

$$\mathbf{E}Y_i = Y_{i+1} = F_i(Y_i) \qquad (i = 0,1,2, \ldots) \tag{2-4}$$

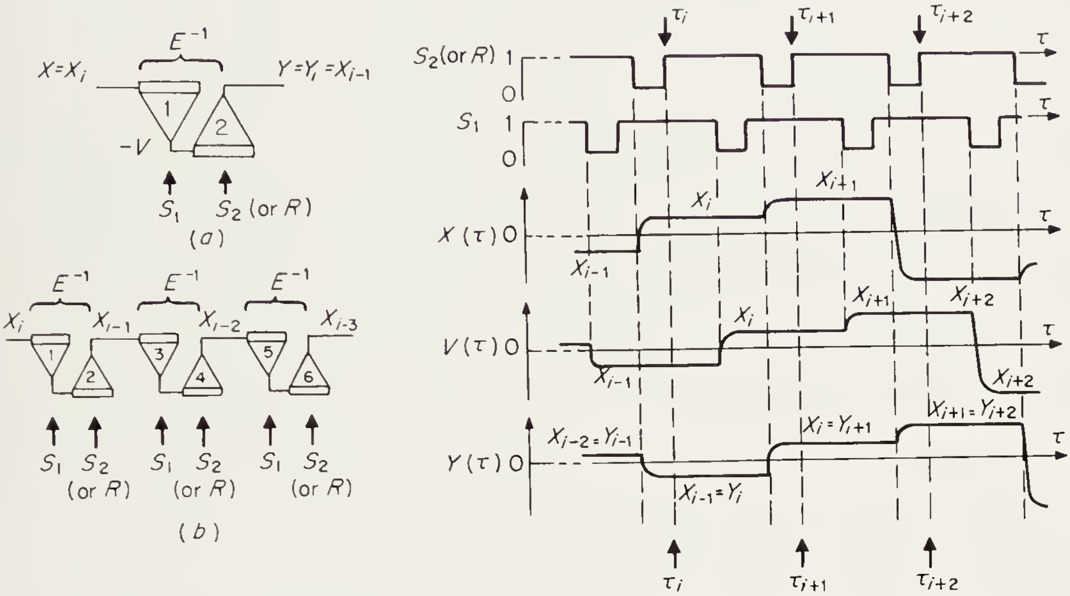


Fig. 2-6. Memory pair with sampled-data input (*a*), and cascaded memory pairs (*b*).

for the unknown data sequence Y_i, given Y_0 (*initial value*) and $F_i(Y_i)$ as a function of Y_i and i. We implement Eq. (4) with a feedback loop (Fig. 2-7*a*) quite analogous to those customarily employed for analog-computer solution of *differential* equations. The circuit of Fig. 2-7*c*, for example, implements the simple difference equation $\mathbf{E}Y_i = aY_i + X_i$. The correct initial memory-pair output Y_0 must be established with an initial-reset circuit (Fig. 2-5*a*).

Figure 2-7*b* similarly solves a *system of first-order difference equations*

$$\mathbf{E}Y_{j,i} = Y_{j,i+1} = F_{j,i}(Y_{1,i};\ Y_{2,i};\ \ldots\ ;Y_{r,i}) \qquad \begin{matrix}(i = 0,1,2, \ldots\ ; \\ j = 1,2, \ldots ,r)\end{matrix} \tag{2-5}$$

for the r unknown sequences $Y_{j,i}\,(j = 1,2, \ldots ,r)$. An *rth-order difference equation*

$$\mathbf{E}^rY_i = Y_{i+r} = F_i(Y_i, Y_{i+1}, \ldots , Y_{i+r-1}) \qquad (i = 0,1,2, \ldots) \tag{2-6}$$

reduces to the form (5) if we introduce Y_{i+1}, Y_{i+2}, . . . as new variables related by the first-order difference equations

$$\mathbf{E}Y_i = Y_{i+1} \qquad \mathbf{E}Y_{i+1} = Y_{i+2} \qquad \cdots \tag{2-7}$$

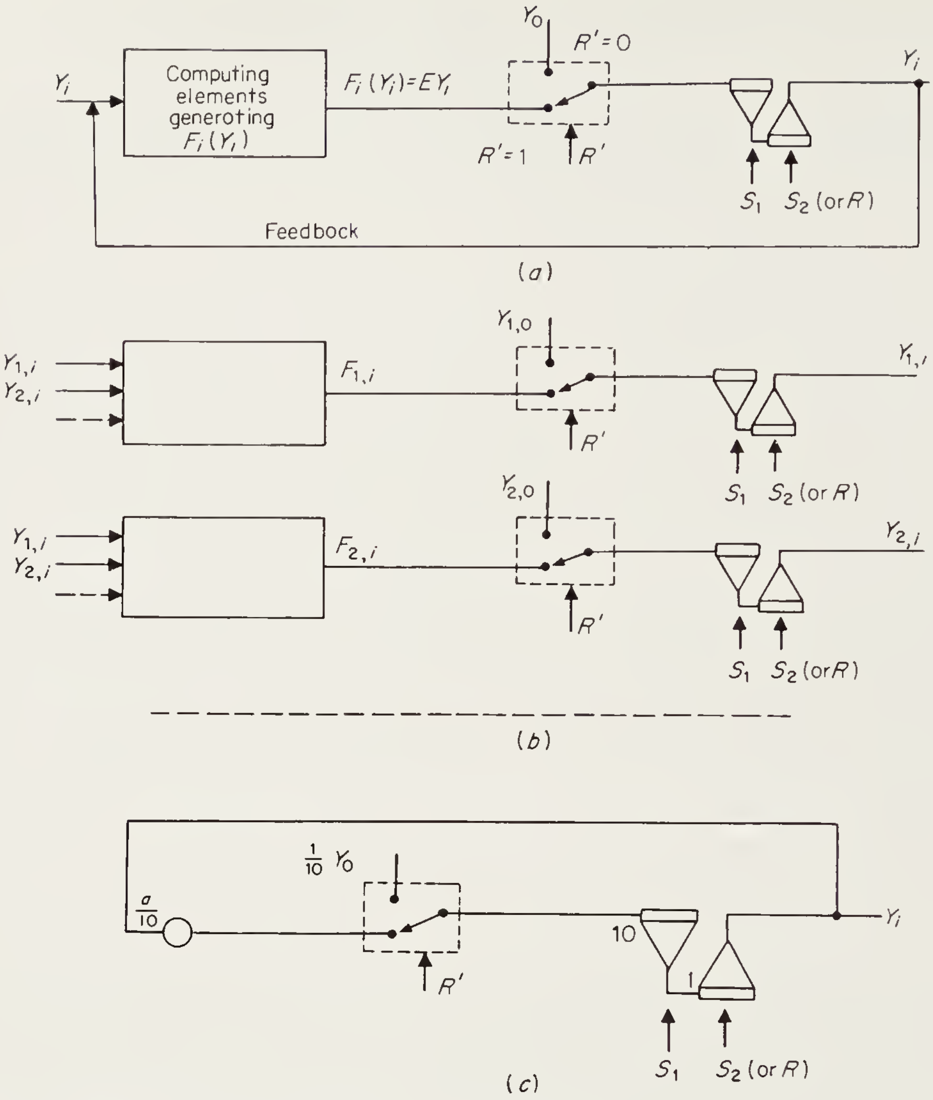


Fig. 2-7. Implementation of recursion formulas (solution of difference equations). The "exponential accumulator" setup in Fig. 2-7*c* implements $EY_i = aY_i$, so that $Y_i = Y_0a^i$ $(i = 0, 1, 2, \ldots)$. This circuit can step parameters for logarithmic-scale plotting.

in exact analogy with Sec. 1-5. Note that the number of initial values given with Eq. (5) or Eq. (6) must again equal the order r of our system; each memory pair requires an initial-reset circuit.

Note that these procedures apply to nonlinear as well as to linear difference equations, and that they do *not* necessarily require periodic sampling, as long as all track-hold pairs are synchronized in the manner of Fig. 2-7*b*. This technique permits analog-computer simulation of relatively complex digital-computer and digital-differential-analyzer operations.

2-4. Linear Operations on Sampled Data. The output sequences Y_i (responses) of many sampled-data processors are related to their input sequences X_i (stimuli) by linear difference equations of the form

$$Y_i + a_{n-1}Y_{i-1} + \cdots + a_0 Y_{i-n} = b_m X_i + b_{m-1}X_{i-1} + \cdots + b_0 X_{i-m} \tag{2-8a}$$

with given constant coefficients a_i, b_k; current-data processing implies $n > m$. Just as in Sec. 1-7, we rewrite the input-output relation (8*a*) in terms of a linear operator $O(\mathrm{E})$ operating on X_i to produce Y_i:

$$Y_i = O(\mathrm{E})X_i = \frac{b_{n-1}\mathrm{E}^{n-1} + b_{n-2}\mathrm{E}^{n-2} + \cdots + b_0}{\mathrm{E}^n + a_{n-1}\mathrm{E}^{n-1} + \cdots + a_0} X_i \tag{2-8b}$$

Such operators commute and may be cascaded and added.

Computer setups implementing the difference equation (8) *are exactly analogous to the computer circuits inplementing the differential equation* (1-11); *it is merely necessary to replace every integrator operation* $-P^{-1}$ *in Figs.* 1-5*a and* 1-5*b by the memory-pair operation* $-\mathrm{E}^{-1}$. Note, however, that memory pairs, unlike integrators, do not usually invert the output-voltage sign.

The difference operator $O(\mathrm{E})$ can also be factored or expressed as a sum of *partial fractions* of the form

$$\frac{1}{A_1\mathrm{E} + A_2}, \quad \frac{A_3\mathrm{E} + 1}{\alpha_0\mathrm{E}^2 + \alpha_1\mathrm{E} + \alpha_3}$$

Partial-fraction representations may reduce switching-spike accumulation, and each partial-fraction term can be implemented and checked separately.

Difference-equation setups conveniently implement relatively complicated sampled-data operations and are especially useful for simulation of digital controllers.

All computer setups should be checked for switching-spike accumulation in memory-pair chains and feedback loops. Reed relays, while slower than electronic switches, are more satisfactory in this respect; a chain of 12 relay track-hold circuits (six memory pairs) can reproduce its input within 25 mV.

HYBRID ANALOG-DIGITAL COMPUTATION

2-5. Different Roles of Digital Computers in Hybrid Computation. We now list the possible jobs of a general-purpose digital computer working with a general-purpose analog computer through a suitable linkage; some of these jobs have been mentioned earlier. The cost of a dedicated small digital computer or of a time-shared larger digital machine is low compared to the engineering costs incurred in simulation studies. Thus, with a reasonable work load, *each one* of the applications listed below will justify the use of the digital computer; and the same digital computer can implement them all.

1. *Supervisory Control of Analog-computer Studies; Storage and Reduction of Results; and Report Preparation.* These jobs define the most important single role of a digital computer in analog/hybrid computation. The digital computer controls the progress of parameter studies by selecting successive parameter combinations, either according to a *predetermined* program or *iteratively in accordance with past results*, as in iterative parameter optimization (Secs. 2-2, 8-14, and 9-13). To complete its supervisory role, the digital computer directs the selection, sampling, and conversion of analog-computer results, performs digital data reduction, plotting, cross-plotting, and printing. It can prepare complete formatted reports.

This type of operation realizes the full benefit of analog-computer speed, while the digital computer, which performs most operations only once per computer run, is not excessively hurried. An especially important application of this kind is the *computation of statistics compiled from thousands of analog-computer runs with random forcing functions, initial conditions, and/or parameters* (*hybrid-computer Monte Carlo studies*). Modern analog computers can produce 100 to 4,000 differential-equation-solving runs with statistically independent random inputs in 1 sec, so that statistically significant results can be quickly obtained. Such high computing speeds, moreover, *give the operator intuitive insight into the effects of parameter changes on statistics*, just as the "slow" differential analyzer provided immediate insight into the results of parameter changes on the solution of a single differential equation.

2. *Digital "Housekeeping" Functions.* Digital-computer lists entered with punched or magnetic tapes can conveniently store the hundreds of coefficient and function-generator settings associated with large simulation studies. The digital computer will not only set these quantities into an analog computer within seconds, but it will also perform automatic *static checks* measuring the propagation of initial conditions throughout the analog-computer setup in the RESET mode (Sec. 7-6) and print errors and diagnostic messages. As a result, substantial "turnaround time" is saved when an old problem is brought back to the analog computer. Some of the newer hybrid computers also permit *digitally controlled patching*, i.e., digital-computer control of the analog interconnections (Sec. 7-4*c*). The digital computer can also implement *diagnostic programs* designed to pinpoint analog-computer malfunctions.

The third and final application area of combined analog/digital computation was historically the first and is the most difficult to use:

3. *Combined Analog-Digital Simulation.* The digital computer operates on some of the problem variables (heretofore only seen in analog form) during a differential-equation-solving run; i.e., the digital computer performs function generation, coordinate transformations, or digital integration.

Combined simulation is motivated by the hope of combining analog-computer speed in some parts of the simulation with digital accuracy and function-generating power in other parts of a simulation. A classical example is analog simulation of an aerospace-vehicle control system and aeroelasticity combined with digital computation of the relatively slowly changing trajectory coordinates (Fig. 2-8*a*). Digital storage and analog output of several complicated functions of two to four variables (frequently required in aerodynamics) is another frequent application of combined-simulation techniques (Fig. 2-8*b* and Sec. 9-9).

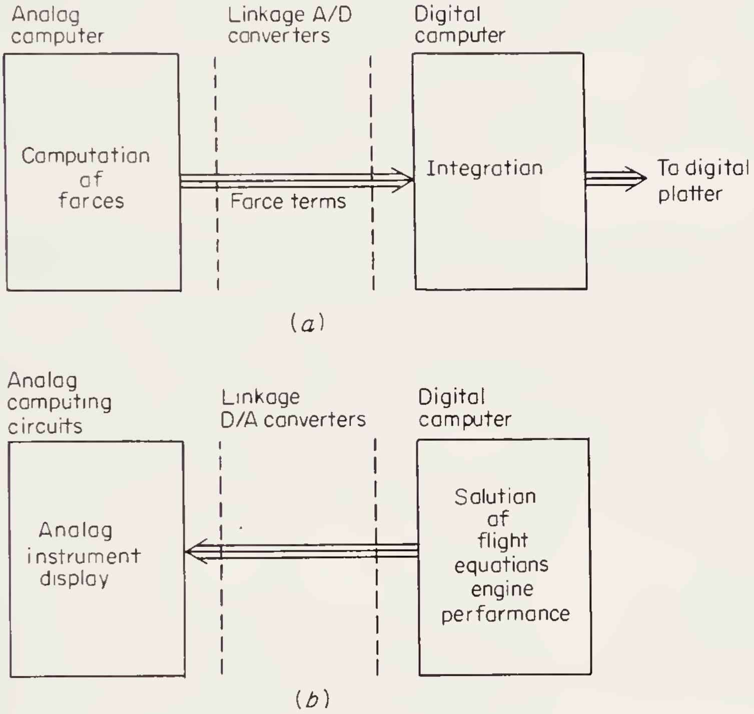


Fig. 2-8. Open-loop combined analog-digital simulation: digital open-loop integration of trajectory equations (*a*) and use of analog instrument outputs with an otherwise all-digital flight simulator (*b*).

Several special cases of combined analog-digital simulations may be distinguished:

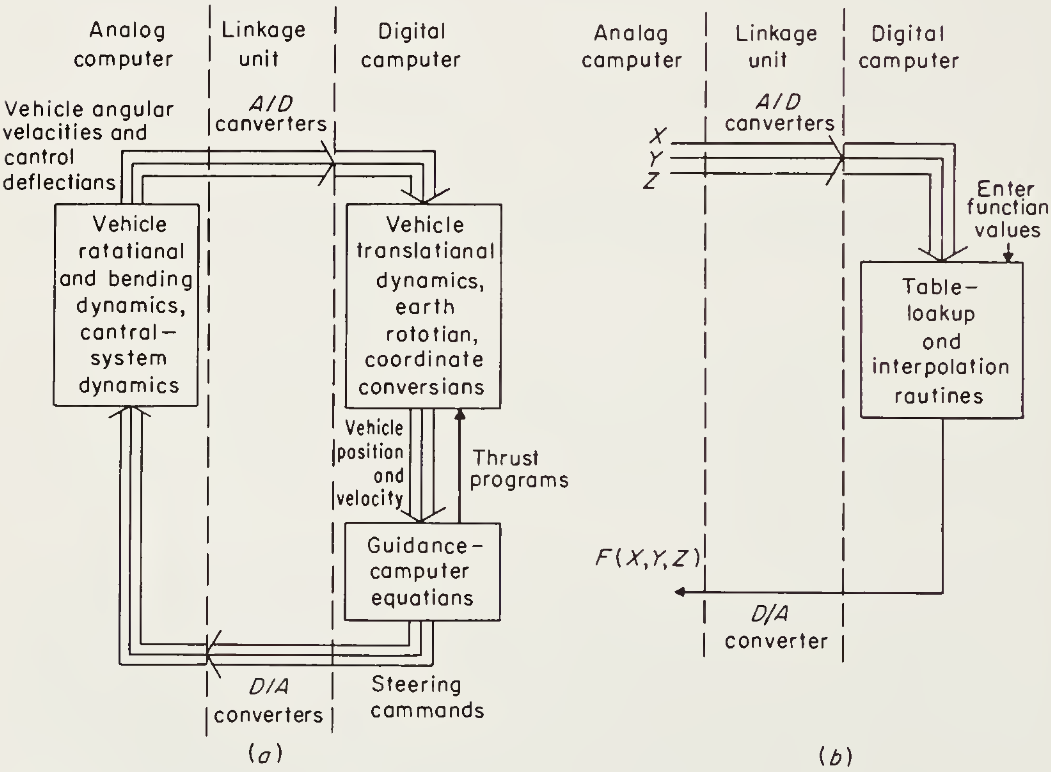
1. *Open-loop Combinations* (Fig. 2-8). (*a*) Digital operations (integration, function generation) on analog-computer outputs without feedback back into the analog computer and (*b*) analog operations on digital-computer outputs without feedback into the digital computer (e.g., instrument outputs for a digital-flight simulator).

These applications are straightforward, subject to the usual speed limitations of digital computers.

2. *Digital-computer Simulation of a Digital Computer.* Here the digital computer operates with analog inputs and outputs in more or less direct analogy with the actual digital controller being modeled.

This, too, is a relatively straightforward problem and will nicely reproduce the performance of the sampled-data control system being simulated.

3. *Closed-loop Combined Simulation.* Here, as in Fig. 2-9, a digital computer not only accepts analog inputs but also produces outputs



Analog computer
Linkage unit
Digital computer
Vehicle angular velocities and control deflections
A/D converters
Vehicle rotational and bending dynamics, control-system dynamics
Vehicle translational dynamics, earth rotation, coordinate conversions
Vehicle position and velocity
Thrust programs
Guidance-computer equations
D/A converters
Steering commands
(a)
Analog computer
Linkage unit
Digital computer
A/D converters
X
Y
Z
Enter function values
Table-lookup and interpolation routines
F(X,Y,Z)
D/A converter
(b)

Fig. 2-9. Closed-loop combined analog-digital simulation: space-vehicle simulation (*a*), and digital function generation (*b*). (*Based on Ref.* 2.)

to the analog computer so that we have a possibly complicated sampled-data feedback loop; some dynamical system variables are modeled in the digital computer.

Applications of this type combine not only the *best* features of analog and digital computation, but also their *worst*, viz., analog-computer inaccuracy and the sampled-data bandwidth limitations of the digital computer. Digital operations imply *sampled-data operations*, *digital processing delays*, and *quantization*, all of which produce essential errors in the representation of continuous dynamical-system variables. These errors must be evaluated and mitigated through careful choice of the portion of the simulation carried out on the digital computer. While quantization errors rarely cause trouble, digital sampled-data operations and processing delays limit the digital part of a combined simulation to slowly varying variables. Even so, subtle loop-instability effects are encountered; it is best to check each problem on a reduced time scale to see whether sampling rates are fast enough for stability (Secs. 9-2 to 9-6).

REFERENCES

1. Korn, G. A.: Project DARE: Differential Analyzer REplacement by On-line Digital Simulation, *Proc. FJCC*, 1969.
2. Goltz, R. A.: The DARE I Digital Simulation System, *Proc. SWIEEECO*, Dallas, Tex., 1970.
3. Liebert, T. A.: On-line Digital Simulation with a Small Computer: The DARE II System, *Proc. Summer Computer Simulation Conference* (ACM, SCi, SHARE sponsored), Denver, Colo., June, 1970.
4. Bekey, G. A., and W. J. Karplus: *Hybrid Computation*, Wiley, New York, 1968.
5. Huskey, H. D., and G. A. Korn: *Computer Handbook*, McGraw-Hill, New York, 1962.

CHAPTER 3

COMPUTER ACCURACY. DESIGN OF NETWORKS, COEFFICIENT POTENTIOMETERS, AND OPERATIONAL AMPLIFIERS

COMPUTER ACCURACY AND OTHER DESIGN SPECIFICATIONS

3-1. A Discussion of Analog-computer Accuracy. Electronic analog computers can produce true solution accuracies varying all the way between 0.01 and 15 percent; and, in each case, such results may or may not be entirely satisfactory. True solution accuracies can, in general, be specified only for special-purpose computers performing a precisely circumscribed class of operations. General-purpose analog computers are applied to a very wide class of problems, and solution accuracies depend so radically on the specific problem solved that one is restricted to the specification of *component accuracies* for individual computing elements. Since some computing-element errors, such as those due to noise and d-c offsets, exist even when the component output voltages are zero, it is customary to specify analog-computer component errors in percent of half-scale (i.e., in volts for ±100-volt analog computers).*

The effects of component accuracies on the computer solution of, say, a system of differential equations are not easy to predict in general and will depend strongly on the problem under investigation. Although a general theory of differential-analyzer errors exists,[1] mathematical error prediction for even moderately complex problems is usually more difficult to obtain than, say, a precise solution of the problem by a digital computer. Hence, many analog-computer centers rely on digital check solutions of every tenth to thirtieth analog-computer solution unless, as is often the case, the investigator can rely on his professional knowledge of the behavior of the simulated systems, components, or subsystems. Checks on known solutions for special parameter values are often helpful (e.g., omission of air resistance and perturbations in ballistic-trajectory computations).

---

* The questionable practice of specifying analog-computer component accuracies in percent of *full-scale* (i.e., of 200 volts for ±100-volt analog computers) is meaningful only after division by 2 and is being gradually discarded.

A perhaps more interesting approach to error estimation is the use of the analog computer itself for the prediction of component-error effects on the final solution. Methods of this type should disclose, for instance, whether or not a given analog-computer solution is "stable" in the sense that small changes in coefficients or component accuracies will not generate large departures from a given solution. Basically, such methods involve the introduction of artificial errors such as bias voltages, phase shift, increased or decreased coefficients, and added step or impulse functions in selected computing elements, and comparison of the resulting changed solutions with the original analog-computer solution.[4] Judicious use of such methods permits an investigator to decide whether suspected or real component inaccuracies can produce serious changes in the analog-computer solution. As an example, a d-c drift voltage added to the output of a simulated servomotor will be naturally decreased if the simulated motor forms part of a simulated high-gain position-servo loop. On the other hand, initial-condition or forcing-function errors in the open-loop integration of a ballistic trajectory may cause substantial errors in the solution. Even more serious error effects are found in solutions of nonlinear oscillation problems, where small changes in initial conditions or coefficients can lead to a mode jump with radically changed solution behavior. In problems of the latter type, no single analog-computer solution can be relied on. It is necessary to show the mode-change behavior through an entire family of solutions with different values of the critical parameter.

More sophisticated versions of this approach have led to various perturbation techniques and to the theory of parameter-influence coefficients (Secs. 8-2 and 8-4). The effects of insufficient computing bandwidth are similarly investigated through repetition of solutions on different time scales (*time-scale check*, Sec. 7-6*c*).

Solution checks by *resubstitution* of computer solutions into the given differential equations have been suggested[2,3] and may become increasingly practical in combined analog-digital computer systems (Sec. 8-5).

3-2. Component Accuracy: Static and Dynamic Errors. (a) Static Accuracy, Step Response, and Small-signal Frequency Response.

Analog computing elements are, in general, most accurate for constant input or inputs (specified *static or d-c component accuracy*) and exhibit increasing *dynamic errors* as the output changes more rapidly. The *response to small and large input-voltage steps* is generally of interest, especially in the case of iterative differential analyzers, where frequent switching is required (Sec. 5-16). Step-response specifications indicate how quickly a computing-element output settles to within a stated percentage of the correct static value after a step disturbance and also specify percent overshoot, ringing, etc.

To describe dynamic errors for small input changes (up to, say, 10 percent of half-scale), we note that a small sinusoidal variation of one of the inputs of operational amplifiers, of multipliers, and of certain function generators will produce a roughly sinusoidal output variation. It is then possible to plot output amplitude and phase for constant input amplitude against frequency (*small-signal frequency response* to the input in question).

Honest component specifications will identify error percentages measured immediately after component adjustments and after a reasonable period of time, say, 8 hr or 1 week. This is especially true for electronic analog computers comprising solid-state components subject to drift with time and/or with temperature changes.

Whenever possible, both static and dynamic accuracy measurements are made by *difference or null methods;* i.e., the difference between actual and desired computing-element outputs for specified inputs is displayed and/or recorded (see Secs. 3-38 and 6-15 for examples).

(b) Phase Errors. The small-signal sinusoidal-error frequency response of many computing elements takes the form shown in Fig. 3-1*a*. We note that *the phase error becomes appreciable long before the amplitude error can even be measured.* Figures 3-1*b* and *c* indicate that such phase-shift errors can cause considerable computing errors in spite of the practically ideal amplitude response; for small errors, the true absolute percentage error due to pure phase error is $\epsilon = 100\delta$, where δ is the phase error measured in radians, or about 1.8 percent per degree of phase error.

For the typical frequency response $a/(j\omega/2\pi f_1 + 1)$(-3 db point at f_1 Hz), the low-frequency phase shift (lag) is $\arctan(f/f_1) \approx f/f_1$ radians at f Hz; *note that we already have* 0.57-*deg or* 1 *percent error at* $0.01f_1$ *Hz.* A slightly underdamped second-order frequency-response function causes less phase shift at very low frequencies but produces step-response overshoot or "ringing."

The usual computing voltages are rarely sinusoidal; typically, their *high-frequency components are smaller in amplitude than low-frequency components.* For instance, in the case of a *symmetrical square wave* with peak-to-peak amplitude $2A$ and period T_0,

$$f(t) = \frac{4}{\pi} A \left(\cos \omega_0 t - \tfrac{1}{3} \cos 3\omega_0 t + \tfrac{1}{5} \cos 5\omega_0 t - \cdots\right)$$

$$\left(\omega_0 = \frac{2\pi}{T_0}\right)$$

the individual sinusoidal components contribute roughly equal absolute phase-shift errors, assuming that phase-shift errors are small and proportional to frequency. In this example, some of the individual phase-shift errors cancel partially; but *a conservative if somewhat arbitrary rule of thumb ensuring adequate step response might be to specify less than* p *deg component phase error for* $50/T$ *Hz, where* p *is the specified static component accuracy in percent of half-scale, and* T *is the duration of a typical computer run.* This rule is, generally speaking, satisfied by the phase inverters of existing analog computers, but less often by their multipliers.

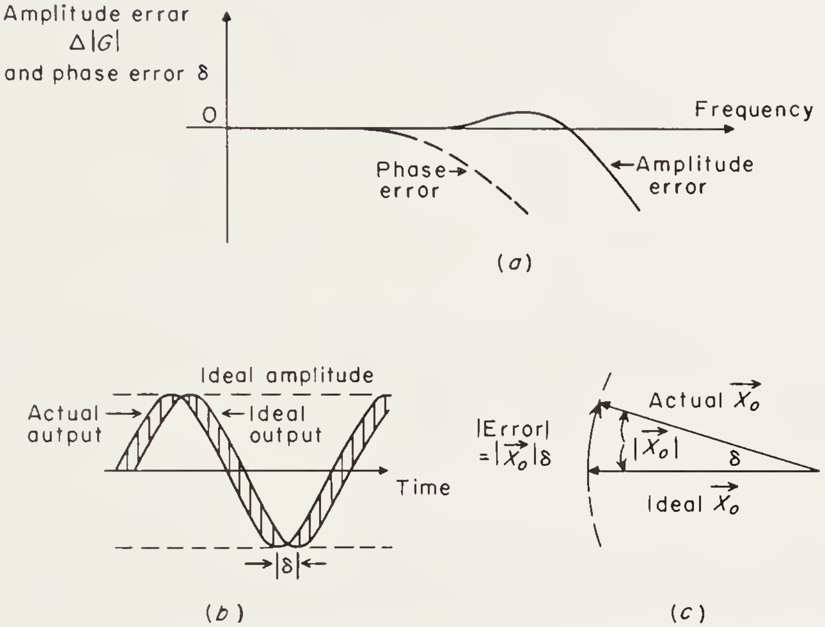


Fig. 3-1. Typical small-signal frequency response of an analog computing element to steady-state sinusoidal input (*a*), and comparison of ideal and actual output sinusoids (*b*) and phasors (*c*). Note that phase shift can cause substantial dynamic errors while the amplitude response is still essentially ideal.

(c) Rate-limit Specifications and Switch-timing Errors. Even with adequate small-signal frequency response, the working-frequency range of most analog computing elements is bounded by *output-voltage rate limiting*, a nonlinear phenomenon usually associated with the currents drawn by load or circuit capacitances, or with servo velocity limiting (Secs. 3-35 and 5-18). A given output-rate limit of, say, r volts/sec necessarily restricts the frequency f of an undistorted sinusoidal output waveform $b \sin 2\pi f\tau$ to $r/2\pi b$ Hz, where b is the sine-wave amplitude. Note, however, that the absolute output-voltage rate of any conventional integrator is never required to exceed $100a$ percent machine units (or volts) per second, where a is the sum of all integrator input gains, since no input can exceed one machine unit in absolute value. This fact determines, for instance, the maximum output rate ever required of a servomultiplier driven by an integrator. Integrator input variables, however, may require much faster rates of change and may, in particular, comprise stepwise changes.

The maximum *timing error* due to a switching delay T_D in integrator mode-control switches or relays, operational relays, or electronic switches and sample-hold readout devices is rT_D, where r is the maximum absolute rate of the voltage being switched. We note that *timing errors rather than computing-element bandwidths may constitute the ultimate speed limitation for fast analog computation.* If, for example, a group of ±100-volt computing elements producing the 10-Hz output

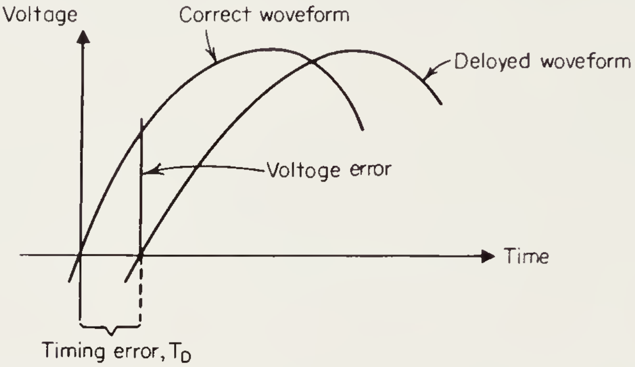


Fig. 3-2. For sinusoidal signal components of frequency f, a time delay T_D causes a true percentage error of $\leqslant L\, 200\pi f T_D$ percent. The delay T_D can be caused by delayed mode-control or signal switching, delayed sampling, delayed comparator response, or digital-computer processing delays (see also Secs. 5-16 and 9-4).

$$X_0 = 100 \sin 2\pi(10\tau) \quad \text{volts} \qquad (\tau > 0) \tag{3-1}$$

are switched into COMPUTE only 0.1 msec too early, the resulting voltage error is $2\pi/10$ volts, or over 0.6 percent of half-scale. Again, in a ±10-volt iterative analog computer producing the 5-KHz waveform

$$X_0 = 10 \sin 2\pi(5{,}000\tau) \quad \text{volts} \qquad (\tau > 0) \tag{3-2}$$

a switching delay of only 0.25 μsec can produce an error of $\pi/40$ volts, or almost 0.8 percent of half-scale (Fig. 3-2).

3-3. Other Design Specifications. Data needed to determine the suitability of computing equipment for a given application (including its compatibility with other system components) comprise specifications such as size, weight, and power-supply requirements, and design limits on temperature, humidity, shock, etc., required to maintain a specified or suitably derated performance.

In the writers' opinion, *maintenance-free operation* (few or no periodic adjustments) ranks in importance with accuracy and frequency response, and frequently ahead of cost. Generally speaking, solid-state devices show smaller aging effects than vacuum tubes but are more sensitive to temperature variations. High-accuracy computing-element designs may place limiter or switching diodes and transistors into small temperature-controlled ovens, employ compensation by temperature-sensitive resistors, or mount solid-state devices in assemblies of high thermal inertia and well away from heat-generating circuits.

COEFFICIENT-SETTING POTENTIOMETERS

3-4. Specifications and Construction (see also Sec. 7-4*b*). A *potentiometer* is an adjustable resistive voltage divider consisting of a fixed resistance with a movable sliding contact. In a *simple coefficient-setting potentiometer* (*scale-factor potentiometer*, *attenuator*), one end of the potentiometer resistance (LO *terminal*) is grounded; the input voltage X_1 is applied to the other end (HI *terminal*), so that the slider (ARM *terminal*) produces the adjustable output voltage $X_0 = \alpha X_1$, where $0 \leq \alpha \leq 1$ (Fig. 3-3*a*). In a *three-terminal coefficient-setting potentiometer*, two input voltages X_1, X_2 are applied to the HI and LO terminals, so that the ARM terminal produces an adjustable output of the form $X_0 = k[\alpha X_1 + (1 - \alpha)X_2]$, where $0 \leq \alpha \leq 1$, $0 < k \leq 1$ (Fig. 3-3*b*; see also Sec. 3-5).

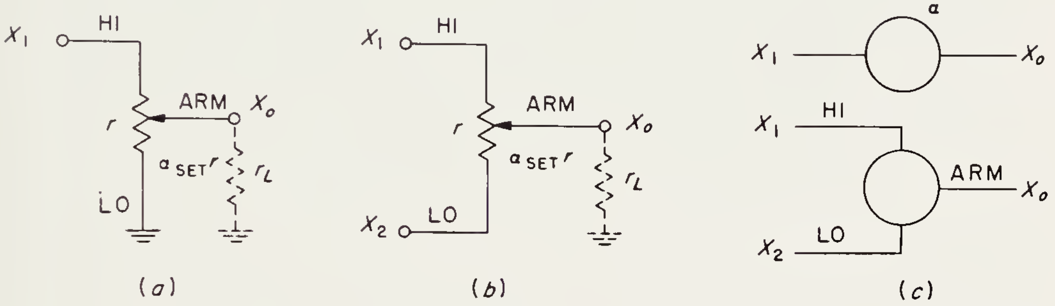


Fig. 3-3. Two-terminal coefficient-setting potentiometer (*a*), three-terminal coefficient-setting potentiometer (*b*), and block-diagram symbols (*c*). The true coefficient α indicated in the block diagram includes the effect of the load (Sec. 3-5). The load resistance r_L is most frequently due to the input resistor of an operational amplifier.

In the most common types of potentiometers, a wirewound resistance wound on a flat card or on a cylindrical mandrel is bent into circular shape, with the sliding contact mounted on a revolving arm. Resistance elements consisting of carbon or metal films deposited on ceramic are also employed. Slider contacts can be spring-mounted precious-metal buttons or bars, or single or multiple wire contacts.[3-6] Potentiometer *resistance* varies between 1,000 and 100,000 ohms and must be determined by a compromise between available amplifier power and potentiometer resolution, which favor high resistance, and resistance stability, phase-shift, and/or loading considerations (Secs. 3-5 and 3-6), which favor lower potentiometer resistance. The resistance of a coefficient-setting potentiometer is usually specified within 0.5 percent and should remain stable within 0.01 to 0.1 percent in a given environment. Most practical coefficient-setting potentiometers exhibit *end resistances* (about 3 ohms at the HI end and 30 ohms at the LO end for a 30K potentiometer), which interfere with precise realization of coefficients equal to one and zero.

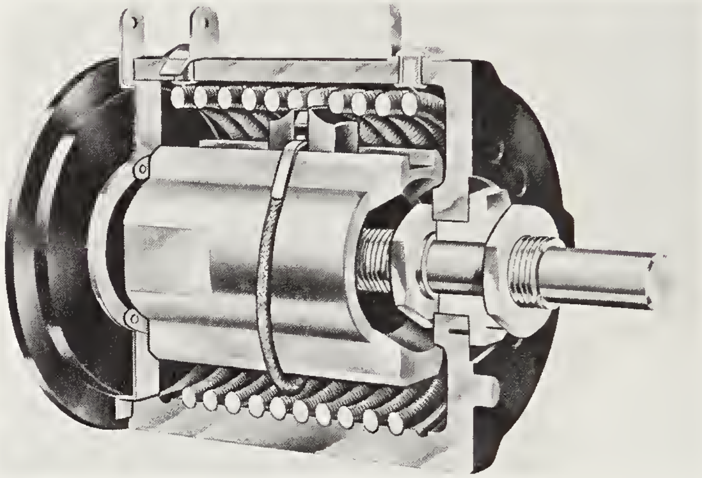


Fig. 3-4. Ten-turn potentiometer with helical resistance element and precious-metal contacts (HELIPOT precision potentiometer). (*Copyright HELIPOT Corporation; reprinted by permission.*)

Computing potentiometers are usually rated between 2 and 5 watts to reduce self-heating effects, even though much less power is actually dissipated. It is wise to fuse the potentiometer-arm connection in order to avoid potentiometer failure due to faulty patching (grounding of the potentiometer arm at high coefficient settings); $\frac{1}{32}$-amp fuses adequately protect 30,000-ohm 5-watt potentiometers, which have current ratings of about 13 mA.

The conformity of the actual coefficient set with a (usually) linear *dial calibration* is of interest in special applications (e.g., comparison potentiometers, Sec. 3-5). In electronic analog computers, however, most coefficient-setting potentiometers are set by actual measurement of their output voltages (Sec. 3-5); so that potentiometer conformity and linearity are not as important as in connection with servo-driven potentiometers used for multiplication and function generation. The static accuracy of a coefficient-setting potentiometer is, then, essentially determined by its *setting stability* (ability to maintain set coefficients), and its *resetting precision*, which involves dial readability and backlash as well as potentiometer setting stability. Both these specifications are quoted in percent of the total potentiometer scale; 0.01 to 0.1 percent are obtainable. Some potentiometer applications require *mechanical ruggedness* to maintain settings in spite of vibrations and shock. In addition, *freedom from microphonism and electrical noise* is desirable.

The resetting precision of a potentiometer cannot be better than its *resolution*. In wirewound potentiometers, resolution is limited by the resistance of one wire turn on a card or mandrel. For increased resolution, the potentiometer-arm contact may slide along a single resistance wire (*slide-wire potentiometer*). More frequently, the resistance wire is wound on a helical mandrel having 2 to 20 turns; so that the potentiometer arm must be capable of axial displacement as well as rotation (*helical potentiometer*, Fig. 3-4). This method of construction is the one most commonly used for coefficient-setting potentiometers and permits better than 0.01 percent resolution for a 30,000-ohm potentiometer. Figure 3-5 shows two types of *multiturn dials* used with such potentiometers.

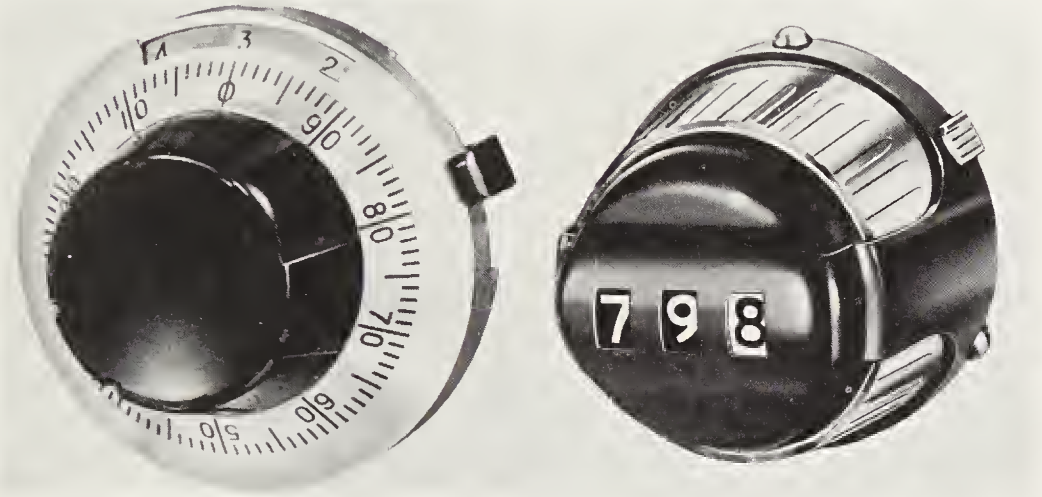


Fig. 3-5. Typical dials for multiple-turn coefficient-setting potentiometers. (*a*) Type RB DUODIAL turn-counting dial (*Copyright HELIPOT Corporation; reprinted by permission.*) (*b*) Borg model 1301 MICRODIAL. (*George W. Borg Corporation.*) Note the dial-lock levers. Such dials can be read and set to 1 part in 10,000 or better, corresponding to the resolution and linearity attainable with high-quality helical potentiometers.

Another type of potentiometer has a rectilinear resistance element with a sliding contact positioned by a lead screw turned by a multiturn dial. *Deposited-film potentiometers* have practically infinite resolution. If the resolution of a given potentiometer is not sufficient for a given application, one may add variable series resistance at the top or bottom of the potentiometer to permit fine adjustment; special dials permit alternate fine and coarse adjustments. Inexpensive computers also use one-turn carbon- or metal-film potentiometers with friction drives; it is difficult to adjust a one-turn potentiometer to within better than about 0.25 percent without some sort of reduction drive.

3-5. Loading Considerations and Potentiometer Setting. Referring to Fig. 3-3*a*, let α_{SET} be the nominal setting or fraction of the total potentiometer resistance r between the ARM terminal and the grounded LO terminal of a potentiometer connected to a load resistance r_L. The node equation for the ARM terminal is readily solved for the output voltage

$$\begin{aligned} X_0 &= \alpha X_1 \\ &= \alpha_{\text{SET}} X_1 \left[1 - \frac{\alpha_{\text{SET}}(1-\alpha_{\text{SET}})(r/r_L)}{1+\alpha_{\text{SET}}(1-\alpha_{\text{SET}})(r/r_L)}\right] \qquad (0 \le \alpha_{\text{SET}} \le 1) \end{aligned} \tag{3-3}$$

The effect of the load tends to make the true coefficient α set on the potentiometer smaller than the nominal setting α_{SET}; specifically,

$$\begin{aligned} \alpha_{\text{SET}} - \alpha &= \frac{\alpha_{\text{SET}}^2(1-\alpha_{\text{SET}})(r/r_L)}{1+\alpha_{\text{SET}}(1-\alpha_{\text{SET}})(r/r_L)} \\ &\left[\approx \alpha_{\text{SET}}^2(1-\alpha_{\text{SET}})\frac{r}{r_L} \approx \alpha^2(1-\alpha)\frac{r}{r_L} \text{ for } \frac{r}{r_L} \ll 1\right] \end{aligned} \tag{3-4}$$

Equation (4) yields exact or approximate *loading corrections* to be added to the desired coefficient α so as to produce the correct nominal potentiometer setting α_{SET} (Fig. 3-6).

In the case of the *three-terminal potentiometer* of Fig. 3-3*b*, the nodal equation for the ARM terminal yields

$$\begin{aligned} X_0 = {} & [\alpha_{\text{SET}} X_1 + (1-\alpha_{\text{SET}})X_2] \\ & \left[1 - \frac{\alpha_{\text{SET}}(1-\alpha_{\text{SET}})(r/r_L)}{1+\alpha_{\text{SET}}(1-\alpha_{\text{SET}})(r/r_L)}\right] \qquad (0 \le \alpha_{\text{SET}} \le 1) \end{aligned} \tag{3-5}$$

Note that the load reduces the output voltage without affecting the ratio of the coefficients of X_1 and X_2.

Potentiometer loading effects prevent the use of linear dial calibrations without loading corrections depending on the specific potentiometer load in each problem. In general-purpose electronic analog computers, one prefers to set coefficients by *direct measurement of the potentiometer output corresponding to a reference-voltage input, with the load (succeeding computing elements) connected;* this does away with the nuisance of loading charts and, incidentally, with the necessity for adjusting a possibly large number of potentiometer-dial zeros accurately. The dial calibration is needed only for rough settings and, possibly, for resetting coefficients used earlier.

Figure 3-7*a* shows how a simple double-pole/double-throw pushbutton switch associated with each potentiometer connects its input to the computer reference power supply and the output to a measuring device (readout bus) without disconnecting the load. The operator can depress the readout button and set the dial with one hand. The simple pushbutton readout system is much more convenient and foolproof than potentiometer-setting schemes employing selector buttons and stepping relays to connect potentiometers to the readout bus; in the writers' opinion, stepping-relay systems are justified only if automatic logging of coefficients by a printer or tape punch is contemplated or if the potentiometers are to be set by an automatic servosystem (Sec. 7-4*b*).

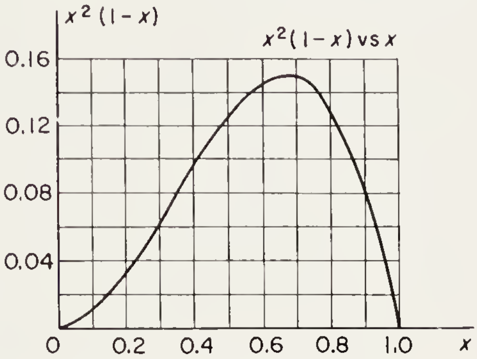


Fig. 3-6. This approximate potentiometer-loading correction chart, based on the relation $\alpha_{\text{SET}} - \alpha \approx \alpha^2(1 - \alpha)(r/r_L)$ $(r/r_L \ll 1)$, is useful for $r/r_L < 0.3$ and setting accuracies down to 0.1 percent. To use the chart, multiply the ordinate corresponding to the desired coefficient α by r/r_L and add the resulting correction to α to obtain α_{SET}. (EXAMPLE: For $r = 20\text{K}$, $r_L = 100\text{K}$, $\alpha = 0.4$, we find $\alpha_{\text{SET}} = 0.4 + 0.02 = 0.42$.) For greater accuracy, one requires separate tables based on the exact formula (3-4) for each value of r/r_L.

Unquestionably, the most convenient voltage-readout device for potentiometer setting is a *digital voltmeter*, which can also serve for setting initial conditions, static checks, and general readout (see also Secs. 7-4 to 7-7). In inexpensive analog computers, potentiometers are set, again with the load connected, by null comparison with a precision *comparison potentiometer* having an accurately calibrated dial (Fig. 3-7*b*). After the desired coefficient is first set on the comparison potentiometer, the operator depresses the potentiometer-readout button and rotates the dial of the coefficient-setting potentiometer until the meter indicates zero current; note that the comparison potentiometer is not loaded at this point. Accurately linear multiturn helical wire-wound potentiometers with counter-type dials (Fig. 3-5*b*) are useful as comparison potentiometers and permit coefficient settings to within better than 0.05 percent.

Very small potentiometer settings (below 0.1) are advantageously set by means of *two* cascaded coefficient-setting potentiometers, with the one closer to the output adjusted first to avoid changes of the load on the first one. *Three-terminal potentiometers* can be set with the LO terminal temporarily grounded, or by an output-voltage measurement depending on the specific application.

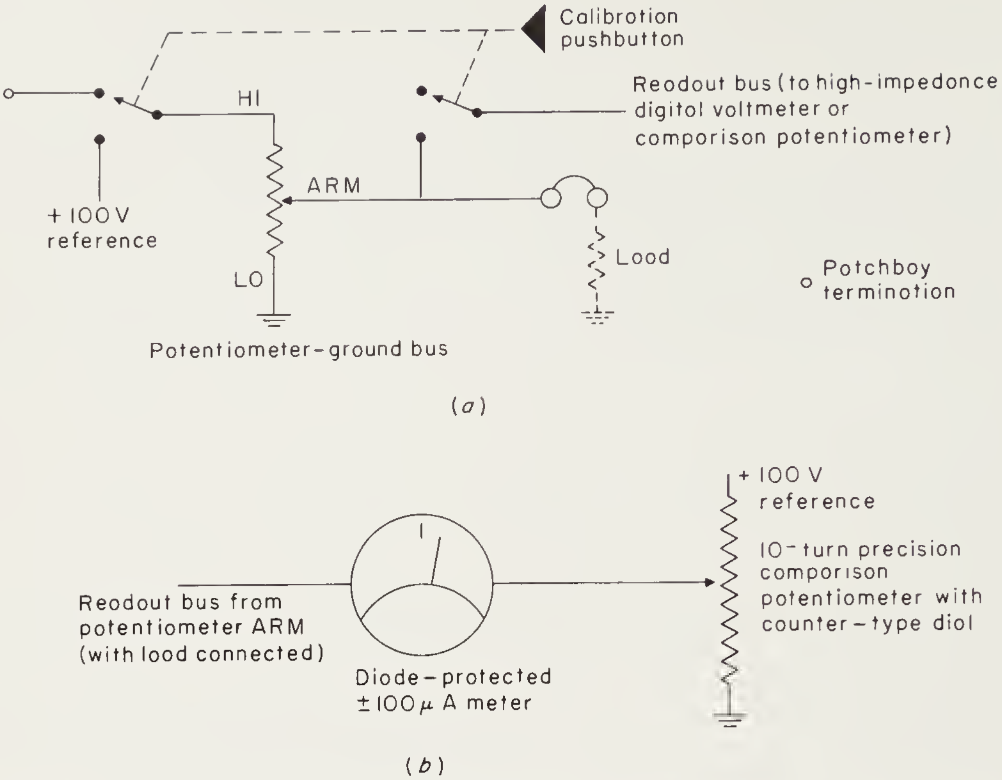

Calibration pushbutton
Readout bus (to high-impedance digital voltmeter or comparison potentiometer)
HI
ARM
LO
+100 V reference
Load
Patchbay termination
Potentiometer-ground bus
(a)
+100 V reference
10-turn precision comparison potentiometer with counter-type dial
Readout bus from potentiometer ARM (with load connected)
Diode-protected ±100 μA meter
(b)

Fig. 3-7. Pushbutton potentiometer-setting circuit (*a*) and comparison potentiometer (*b*). It is customary to disconnect and ground (or feedback-ground) all amplifier summing junctions during the potentiometer-setting operation to prevent amplifier overloads (POTSET mode, Sec. 7-3*a*).

Servo setting of coefficient-setting potentiometers and digital attenuators will be discussed in connection with semiautomatic and automatic programming systems in Sec. 7-4*b*.

3-6. Dynamic Errors and Capacitor Compensation. With increasing computing frequencies, the high-frequency phase shift in coefficient-setting potentiometers may greatly exceed phase shift in other computing elements and can introduce serious dynamic errors into computations (see also Sec. 3-24). Potentiometer phase shift is mainly due to the capacitive load produced by computer circuits and wiring, and by the distributed capacitances of the potentiometer resistance, housing, and/or copper mandrel. With wirewound potentiometer resistance elements wound on flat cards or small-diameter mandrels, the effects of potentiometer *inductance* are not ordinarily felt at computing frequencies below 1 MHz.

Potentiometer and load capacitances cause both lag and lead effects. Figure 3-8 shows a simplified equivalent circuit which matches the frequency-response characteristics of helical copper-mandrel potentiometers quite accurately;[9,13,14] Fig. 3-9 shows typical phase shift vs.

frequency for a 30K potentiometer of this type. While an ideal voltage divider would not introduce any phase changes at all, the wirewound potentiometer is seen to introduce a lag at high potentiometer settings and a lead at low potentiometer settings.

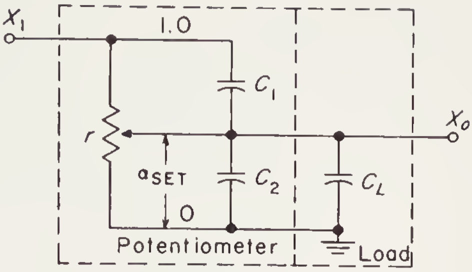


Fig. 3-8. Simplified equivalent circuit for a 10-turn copper-mandrel potentiometer (Beckman Helipot), useful for potentiometer settings α_{SET} between 0.05 and 0.95, r between 10 and 100K, and frequencies below $1/24\pi r C_1$. For type A Helipots, C_1 equals C_2 and varies between 200 pF ($r = 10$K) and 225 pF ($r = 100$K). C_L is the load capacitance, including wiring capacitance. The potentiometer phase shift is

$$\delta_P = \omega r C_1 (1 - \alpha_{SET}) \left(1 - \alpha_{SET} \frac{C_1 + C_2 + C_L}{C_1}\right)$$

(*Based on Ref.* 14.)

.Potentiometer phase shift decreases with the potentiometer resistance, but this cannot be reduced below a minimum value determined by amplifier power and, in wirewound potentiometers, by the required resolution. For fast (repetitive) analog computers which do not require setting stability below 0.1 percent, the best type of coefficient-setting potentiometer might be a type with a linear deposited-film resistance element with the slider positioned by a multiturn dial turning a lead screw. Such potentiometers have negligible inductance and distributed capacitance; note, though, that potentiometer construction can do nothing about the effects of capacitive loading due to computer circuits and wiring.

Capacitor compensation permits the use of high-stability wirewound helical potentiometers at frequencies as high as 20 KHz. Suitably

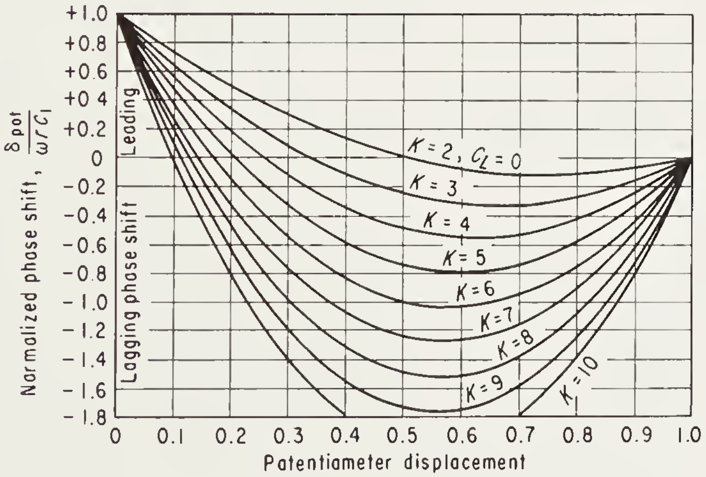


Fig. 3-9. Normalized potentiometer phase shift $\delta_P/\omega r C_1 \approx (1 - \alpha_{SET})(1 - K\alpha_{SET})$, where $K = (C_1 + C_2 + C_L)/C_1 = 2 + C_L/C_1$ and $r_L \gg \alpha_{ST}\ (1 - \alpha_{SET})r$. The approximate formula for $\delta_P/\omega r C_1$ is valid up to 6.6 KHz for $r = 10$K and up to 600 Hz for $r = 100$K. (*From Ref.* 14.)

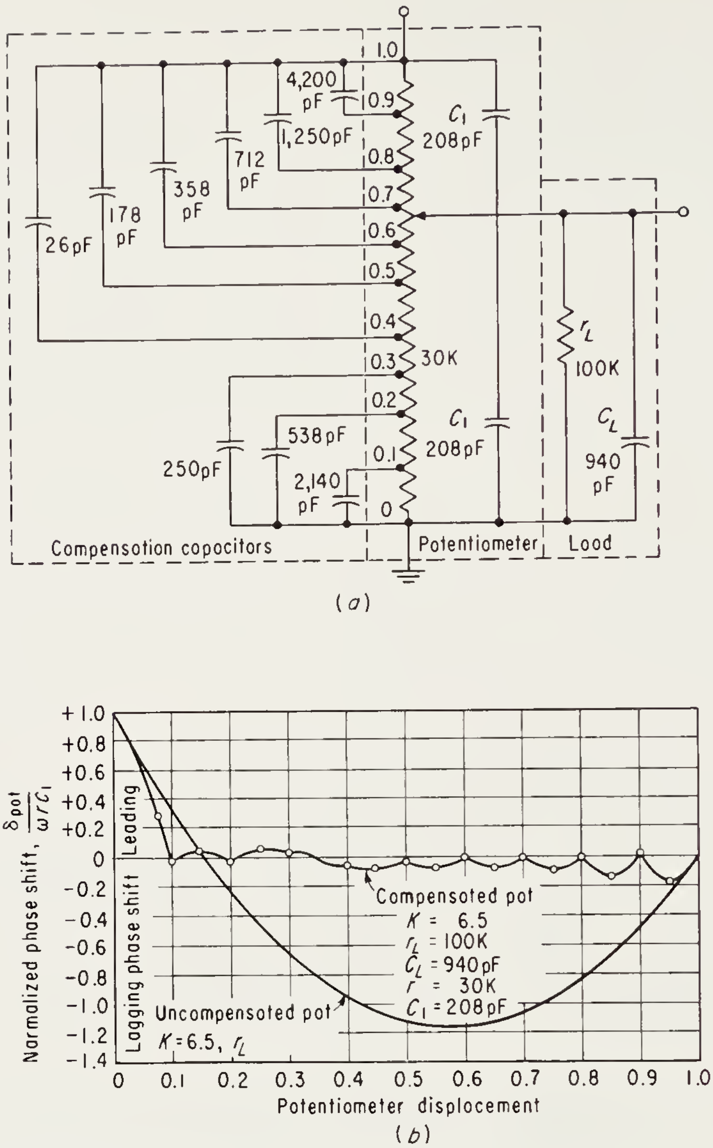


Fig. 3-10*a*, *b*. Phase compensation of a potentiometer (*a*) and effect on phase shift (*b*). Phase compensation is designed for a specific resistance-capacitance load.[14]

chosen compensating capacitors are switched, patched, or plugged between the ARM and HI terminals for high potentiometer settings, and between the ARM and LO terminals for low settings. The correct capacitance values for different potentiometer settings, load resistances, and load capacitances may be tabulated for each type of potentiometer. One can also select compensating capacitors with the aid of an oscilloscope displaying the potentiometer output for a square-wave test input; conceivably, such test circuits could be combined with the potentiometer-setting system.

If the resistive and capacitive loads on the coefficient-setting potentiometers are known or can be standardized, compensating capacitors

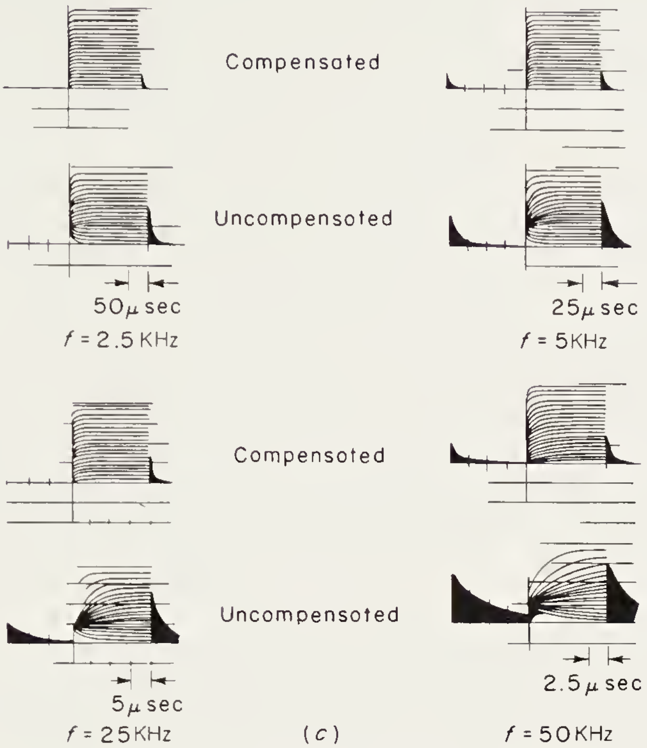


Fig. 3-10*c*. Effect of phase compensation on potentiometer step response.[14]

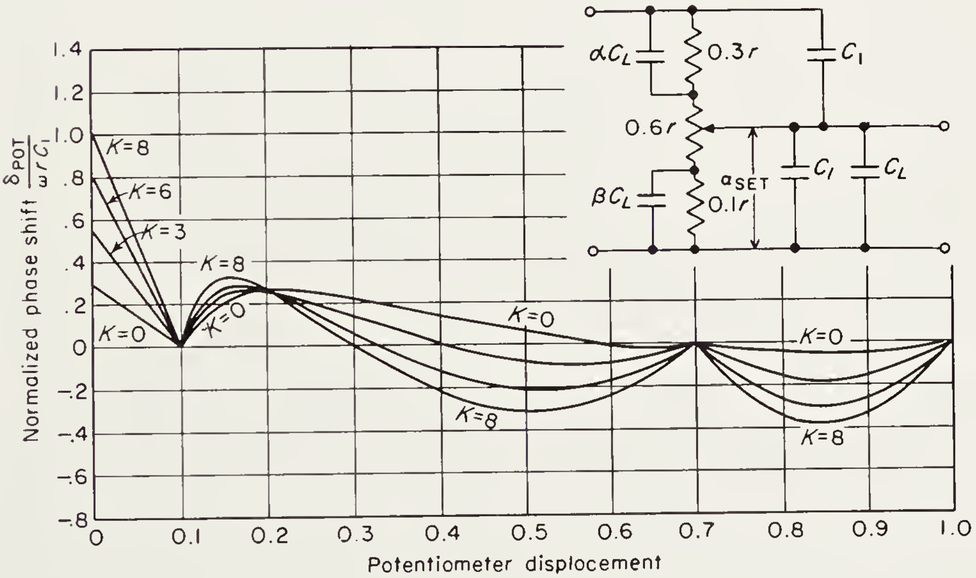


Fig. 3-11. This simple 2-tap compensation circuit compares nicely with the 10-tap circuit of Fig. 3-10; note the nonuniform tap spacing.[13]

could be switched simply in accordance with the potentiometer setting. As a perhaps more practical alternative, Figs. 3-10 and 3-11 show the *application of compensating capacitors to potentiometer taps* and illustrate the remarkable frequency-response improvement obtainable by this method.[8,9,13,14]

COMPUTING RESISTORS AND CAPACITORS. PARASITIC-IMPEDANCE EFFECTS

3-7. Computing Resistors. (a) Resistance Stability. Wirewound Computing Resistors. *Resistance stability* is the ability of a resistor to maintain its nominal resistance value. Note that many committed summing amplifiers and electronic multipliers require accurate resistance *ratios*, which are easier to establish and maintain than individually accurate resistance values. References 10 to 12 describe detailed practical methods for precision measurement or matching of resistances or resistance ratios.

Most computing resistors are *wirewound resistors* noninductively wound on ceramic or thermosetting-plastic rods. The resistance winding undergoes a temperature- and current-recycling process to eliminate instability (relaxation effects due to stresses in the wire). The initially intentionally high resistance is then trimmed to its nominal value. To obtain a stable resistor, one must carefully maintain the insulation between windings, which are usually separated into individual subwindings or pies; prolonged baking and immediate impregnation or encapsulation with plastic or ceramic keeps out moisture.

Wirewound resistors stable to within 50 to 1,000 ppm at constant temperature are commercially available; and suitable resistance-wire alloys yield *temperature coefficients* less than 10 ppm/deg C. Resistors should, of course, be trimmed at or near their typical operating temperature. One attempts to minimize the effects of *self-heating* due to the current in the resistor by using the largest power rating (wire diameter and radiating surface) practical for a given physical size, and by providing suitable air circulation. Resistance changes due to self-heating may vary between 3 and 3,000 ppm/watt at rated current and constant ambient temperature. Plastic encapsulation tends to increase the thermal inertia of a resistor, but may aggravate self-heating effects. Resistor installation should also guard against generation of spurious voltages due to thermal gradients and contact potentials.

Constant-temperature ovens used with some larger analog computers maintain resistance values within ±15 ppm for ambient temperature changes of ±20 deg C (see also Sec. 3-8). Resistors for electronic analog computers operating in a reasonable indoor environment will, in the writers' opinion, not really require constant-temperature ovens; but such ovens are often provided in any case for computing capacitors, and the use of an oven may permit the designer to specify slightly less expensive resistance wire.

Resistors used to make up resistance ratios are often encapsulated together in plastic or installed in a common oil bath, so that some temperature effects are effectively canceled out.

(b) Parasitic Inductance and Capacitance. Film Resistors. Figure 3-12 shows a simple a-c equivalent circuit for a physical resistor. The equivalent impedance $Z_R(j\omega)$ of the resistor is given by

$$Z_R(j\omega) = R_{DC}\frac{j\omega(L_R/R_{DC}) + 1}{j\omega R_{DC}C_R + 1 - \omega^2 L_R C_R} \quad (3\text{-}6)$$

where $R_{DC} = Z_R(0)$ is the nominal (d-c) resistance value. The physical resistor effectively constitutes a low-Q resonant circuit, which can be tested by pulse excitation.

Wirewound computing resistors should have noninductive windings, not just to minimize the parasitic inductance L_R but also to reduce inductive pickup and crosstalk at high frequencies. For noninductively wound wirewound resistors, the values of parasitic inductance L_R will be below $100R_{DC}$ mH, if R_{DC} is measured in megohms. For $R_{DC} > 25$K, the effect of the inductance L_R is essentially negligible at signal frequencies below 1 MHz, in which case

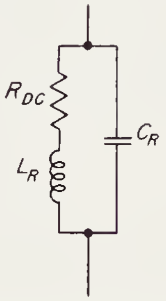


Fig. 3-12. High-frequency equivalent circuit for a wirewound resistor.

$$Z_R(j\omega) \approx \frac{R_{DC}}{j\omega R_{DC}C_R + 1} \quad (3\text{-}7)$$

Typical values of the resistor distributed capacitance C_R range between $1/2R_{DC}$ and $4/R_{DC}$ pF, if R_{DC} is measured in megohms. Resistor distributed capacitance has no adverse effect in properly designed summing amplifiers at computing frequencies up to 1 MHz (Sec. 3-19). In integrators, however, errors due to resistor distributed capacitance prevent the use of wirewound resistors greater than 10^5 ohms at about 100 Hz (Sec. 3-20).[11,12]

Fast computing circuits employ *metal-film resistors* with practically negligible inductances (0.01 μH) and distributed capacitances essentially independent of resistance; 0.5 pF is a typical end-to-end capacitance for a ½-watt film resistor. Accurate circuits require *thin-film* resistors (evaporated on a ceramic substrate), which can have temperature coefficients as low as ±50 ppm/deg C (−70 to +150 deg C) and stability within ±30 ppm/year. Note that *different metal-film resistors (especially with different orders of magnitude of resistance) will, in* *general, have different positive or negative temperature coefficients.* Selection can be used for temperature compensation of resistor networks. Resistance *ratios* obtained with resistances of approximately similar magnitude can track within 1 ppm/deg C (0 to 60 deg C). Thin-film elements are also recommended for *variable-resistance trimmers.*

Thick-film or cermet resistors (metal sprayed on a ceramic substrate) are used in circuits requiring less accuracy (250 ppm/deg C; at most 0.1 percent of half-scale), and especially in hybrid integrated circuits. Complete resistance networks can be either evaporated or sprayed on a common substrate to improve temperature tracking and to reduce manufacturing costs (see also Sec. 5-7). *Carbon film* has a relatively large negative temperature coefficient (up to 500 ppm/deg C) and is recommended only for temperature compensation.

3-8. Computing Capacitors: Stability. The nominal capacitance value for a given capacitor depends somewhat on the method of measurement (Sec. 3-9). Assuming that a capacitance-measurement method suitable for definition of computing capacitances is agreed on, one finds that capacitance values not only vary with temperature but are affected by their recent history with respect to temperature changes and mechanical stresses. Table 3-1 lists temperature coefficients of capacitance for various types of capacitors. If one desires to maintain nominal capacitance value within 0.05 percent, a constant-temperature oven, maintained at a temperature somewhat above room temperature, may be required; a fast-cycling oven will stabilize capacitor temperatures within

Table 3-1. Properties of Dielectrics Used for Computing Capacitors (Typical Values)

Dielectric	Dissipation coefficient D (at 10 Hz and 25 deg C)	Insulation resistance r_{DC} (megohms/μF)	Temperature coefficient of capacitance
Mylar.......	8×10^{-4} to 14×10^{-4}	2×10^5 (20 deg C) 10^3 (100 deg C)	+250 ppm/deg C (20 to 80 deg C; larger at high and low temperatures)
Polystyrene...	1.4×10^{-4} to 2×10^{-4} (decreases from this figure with increasing frequency up to about 70 KHz, then increases again)	2×10^6 (20 deg C) 2×10^5 (60 deg C)	−50 to −100 ppm/deg C (−60 to +60 deg C)
Teflon	0.7×10^{-4} to 1.2×10^{-4}	10^6 (0 deg C) 100 (160 deg C)	−250 ppm/deg C (−60 to +160 deg C)

a narrower range than a slow-cycling oven.[12] In addition, any appreciable temperature transient will necessitate capacitor readjustment (*temperature-retrace error*).[14] Retrace accuracy for one to five temperature cycles between −70 and +120 deg C can be within 0.03 percent for high-quality plastic-film capacitors, with a maximum capacitance change below 0.1 percent. Adjustment of capacitors located in an oven must be made without temperature transients after adjustment. It may be best to locate trimmer capacitors outside the capacitor oven. Ovens *refrigerated* to just below room temperature are also an attractive possibility.

3-9. Capacitor Leakage and Dielectric Absorption (see also Secs. 3-20 and 3-24).[96] **(a) Resistive Leakage.** The effects of *resistive leakage* shunting a computing capacitance can be reduced completely to leakage in the capacitor dielectric proper through installation of grounded conductors surrounding all capacitance terminations ("insulation with metal"; see also Sec. 3-20). The remaining leakage resistance r_{DC} in high-quality polystyrene capacitors will exceed 5×10^5 M/μF at 100 deg F.* The time constant of the resulting parallel resistance-capacitance combination will, then, exceed 5×10^5 sec, so that purely resistive leakage will affect computer accuracy noticeably only for relatively long computing or holding times or low frequencies (see also Fig. 3-13).

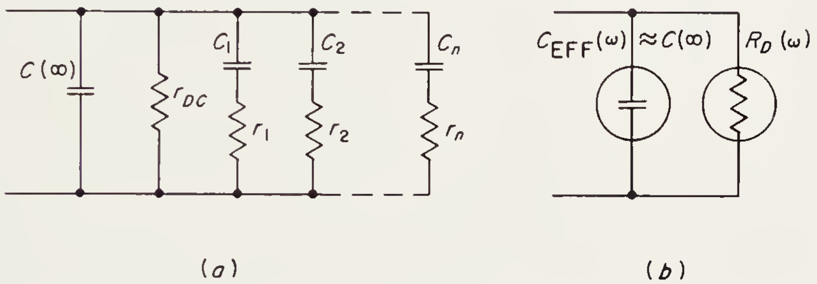


Fig. 3-13. Linear capacitor model (*a*) and nonlinear model (*b*). For a typical 1-μF 200-volt polystyrene computing capacitor[13,14] $C(\infty) = 1\ \mu$F. $r_{DC} = 10^6$ M with the r_k, C_k combinations 60 pF, 2.65×10^6 M; 132 pF, 1.21×10^5 M; 204 pF, 7.80×10^3 M; 216 pF, 737 M; 219 pF, 72.7 M; 234 pF, 6.80 M. The resulting linear model is valid between 0 and 100 Hz.

(b) Dielectric Absorption. Capacitor Models. Unfortunately, resistive leakage does not cause the only energy loss in capacitors. Activation-energy thresholds delay the dielectric polarization associated with capacitor-voltage changes and thus cause frequency-dependent energy dissipation commonly referred to as *capacitor dielectric absorption.* This "relaxation" effect can be represented to a good approximation by a set of fixed leakage resistances $r_1, r_2, \ldots, r_n$ respectively in series with small capacitances $C_1, C_2, \ldots, C_n$ (Dow's capacitor model, Fig. 3-13*a*); values of n between 1 and 10 have been used.[10,13–18] The capacitor impedance operator given by the equivalent circuit of Fig. 3-13*a* can still be written in the form $Z_C(P) = 1/C(P)P$, where the "capacitance" $C(P)$ is no longer a constant but an operator

---

* Leakage tends to increase with temperature. Where capacitor ovens are used, their temperatures should not greatly exceed room temperature; 100 deg F is a reasonable value.

$$C(P) = C(\infty) + \frac{1}{r_{DC}P} + \sum_{k=1}^{n} \frac{C_k}{1 + r_k C_k P} \tag{3-8}$$

For steady-state sinusoidal voltages and currents, one has the complex impedance $Z_C(j\omega) = 1/j\omega C(j\omega)$, where $C(j\omega)$ is the *complex capacitance*

$$\begin{aligned} C(j\omega) &= C(\infty) + \frac{1}{j\omega r_{DC}} + \sum_{k=1}^{n} \frac{C_k}{1 + j\omega r_k C_k} \\ &= C(\infty) + \frac{1}{j\omega r_{DC}} + \sum_{k=1}^{n} \frac{1 - j\omega r_k C_k}{1 + \omega^2 r_k^2 C_k^2} C_k = |C(j\omega)| e^{-jD(\omega)} \end{aligned} \tag{3-9}$$

where $$D(\omega) = -\arg C(j\omega) = \frac{1}{2\pi} \frac{\text{energy loss per cycle}}{\text{peak energy stored}} \tag{3-10}$$

is the *capacitor dissipation coefficient*, which measures both dielectric absorption and resistive leakage. In properly installed polystyrene capacitors, resistive leakage is negligible compared with dielectric absorption above 0.001 Hz, so that $D(\omega)$ is essentially identical with the *coefficient of dielectic absorption* at higher frequencies.

One may also write the admittance $Y_C(j\omega) = 1/Z_C(j\omega)$ of the physical capacitor as

$$Y_C(j\omega) = \frac{1}{Z_C(j\omega)} = j\omega C(j\omega) = \frac{1}{R_D(\omega)} + j\omega C_{\text{EFF}}(\omega) \tag{3-11}$$

in terms of the real but frequency-dependent resistance $R_D(\omega)$ and capacitance $C_{\text{EFF}}(\omega)$ of an *equivalent resistance-capacitance shunt circuit* (Fig. 3-13*b*), where

$$\left.\begin{aligned} \frac{1}{R_D(\omega)} &= -\omega \operatorname{Im} C(j\omega) = \frac{1}{r_{DC}} + \sum_{k=1}^{n} \frac{\omega^2 r_k C_k^2}{1 + \omega^2 r_k^2 C_k^2} \\ C_{\text{EFF}}(\omega) &= \operatorname{Re} C(j\omega) = C(\infty) + \sum_{k=1}^{n} \frac{C_k}{1 + \omega^2 r_k^2 C_k^2} \end{aligned}\right\} \tag{3-12}$$

so that

$$C(j\omega) = C_{\text{EFF}}(\omega) + \frac{1}{j\omega R_D(\omega)} = C_{\text{EFF}}(\omega)[1 - jD(\omega)] \tag{3-13}$$

with $$\begin{aligned} D(\omega) &= -\arg C(j\omega) = \frac{1}{\omega R_D(\omega) C_{\text{EFF}}(\omega)} \\ &= \frac{1}{\omega r_{DC} C_{\text{EFF}}(\omega)} + \sum_{k=1}^{n} \frac{\omega r_k C_k}{1 + \omega^2 r_k^2 C_k^2} \frac{C_k}{C_{\text{EFF}}(\omega)} \end{aligned} \tag{3-14}$$

$C_{\mathrm{EFF}}(\omega)$ is the "effective" capacitance actually measured by bridge or oscillator test circuits with sinusoidal excitation at the circular frequency ω; note that $C_{\mathrm{EFF}}(\omega)$ is a function of frequency. For all high-quality capacitors, $\sum_{k=1}^{n} C_k \ll C(\infty)$, so that $C_{\mathrm{EFF}}(\omega) \approx C(\infty)$ varies only slightly with frequency (about -0.02 percent per decade of frequency increase for polystyrene between 0.1 and 100 Hz).

With suitably chosen parameters $C(\infty)$, r_{DC}, C_k, r_k, Dow's capacitor model matches actual phase-shift measurements very accurately (Fig. 3-14). For practical dielectrics, one finds that the dissipation coefficient $D(\omega)$ is practically constant between 0.1 and 1,000 Hz. Table 3-1 lists dissipation coefficients for various dielectrics; dielectric absorption in actual plastic films will depend mainly on the dielectric grains and impurities obtained in each specific manufacturing process.[15,18] Polystyrene capacitors are most commonly used in high-quality electronic analog computers. Production of improved polyethylene and Teflon films is still experimental but holds promise for the future. Mylar is a reasonable substitute for polystyrene in low-cost applications.

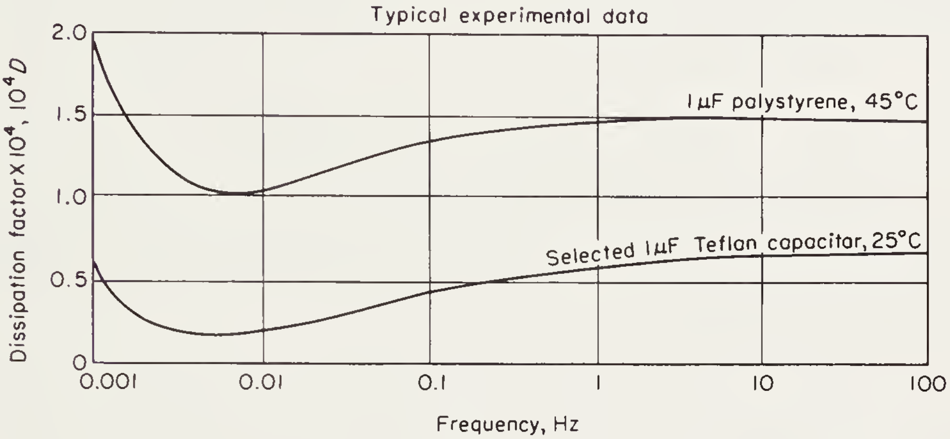


Fig. 3-14. Capacitor dielectric absorption vs. frequency for polystyrene and Teflon capacitors. (*From Ref.* 14.)

(c) Effects of Dielectric Absorption.[96] The capacitor model of Fig. 3-13*a* correctly accounts for the following effects of dielectric absorption in physical capacitors:

1. Sinusoidal current leads voltage by $\pi/2 - D(\omega)$ radians, as compared with the $\pi/2$ lead angle of ideal capacitors.
2. An initially charged and discharged capacitor will gradually recover a fraction of its earlier charge; the linear model reproduces this effect as the various capacitor charges redistribute themselves in a manner depending on the past charging history ("*soakage*" or *memory effect* of dielectric absorption).

3. Results of accurate capacitance measurements will depend on the method of measurement. In particular, the results of bridge and oscillator measurements depend on the frequency components of the test signals used (see also Sec. 3-20); and capacitance measurements based on timed charging or discharging are affected by the past charging history of the capacitor under test.[10]

For purposes of analog computation, Single[14] suggests successive adjustment of three similar integrating capacitors in an accurately timed computer setup for $d^2x/dt^2 = -\omega^2 x$ (Fig. 1-4*d*) with accurately known computing resistors; ω^2 is a power of 10 chosen so as to yield a sinusoidal frequency $\omega/2\pi$ near the center of the computer working-frequency range. $\omega^2 = 1$ would be a suitable choice for a "slow" electronic analog computer.

THE PARALLEL-FEEDBACK OPERATIONAL AMPLIFIER

3-10. Introduction. The general parallel-feedback operational-amplifier circuit of Fig. 3-15 is usually designed with the aid of the "ideal" performance equation

$$X_0 = -Z_0(P)\left[\frac{X_1}{Z_1(P)} + \frac{X_2}{Z_2(P)} + \cdots + \frac{X_n}{Z_n(P)}\right] \qquad \left(P \equiv \frac{d}{d\tau}\right) \qquad (3\text{-}15)$$

already derived in Sec. 1-10 on the assumption of infinite loop gain. Here, $Z_0, Z_1, Z_2, \ldots, Z_n$ are short-circuit transfer impedances computed for each four-terminal network as the operational ratio between input voltage and short-circuit output current, as in Fig. 1-8; expressions for the transfer impedances of many networks are available in published tables like Table A-1.

The performance equation (15) neglects d-c offset and noise effects and holds, in any case, exactly only if the feedback loop gain is sufficiently large to make the summing-point voltage E_G practically negligible. In this case, Eq. (15) follows from the summing-point node equation. It is the purpose of the following sections to derive more exact performance equations, which will permit the designer to estimate errors due to the use of Eq. (15) in the more realistic case of finite loop gain. The effect of the output load on the loop gain will also be investigated. In addition, Secs. 3-15 and 3-16, respectively, treat d-c offset and noise effects and the effects of gain changes and distortion. Errors due to parasitic impedances in the computing networks are discussed in Secs. 3-18 to 3-21.

3-11. Operational-amplifier Performance with Finite Gain and Loading.[19] Since any output due to d-c offset and noise in Fig. 3-15*a*

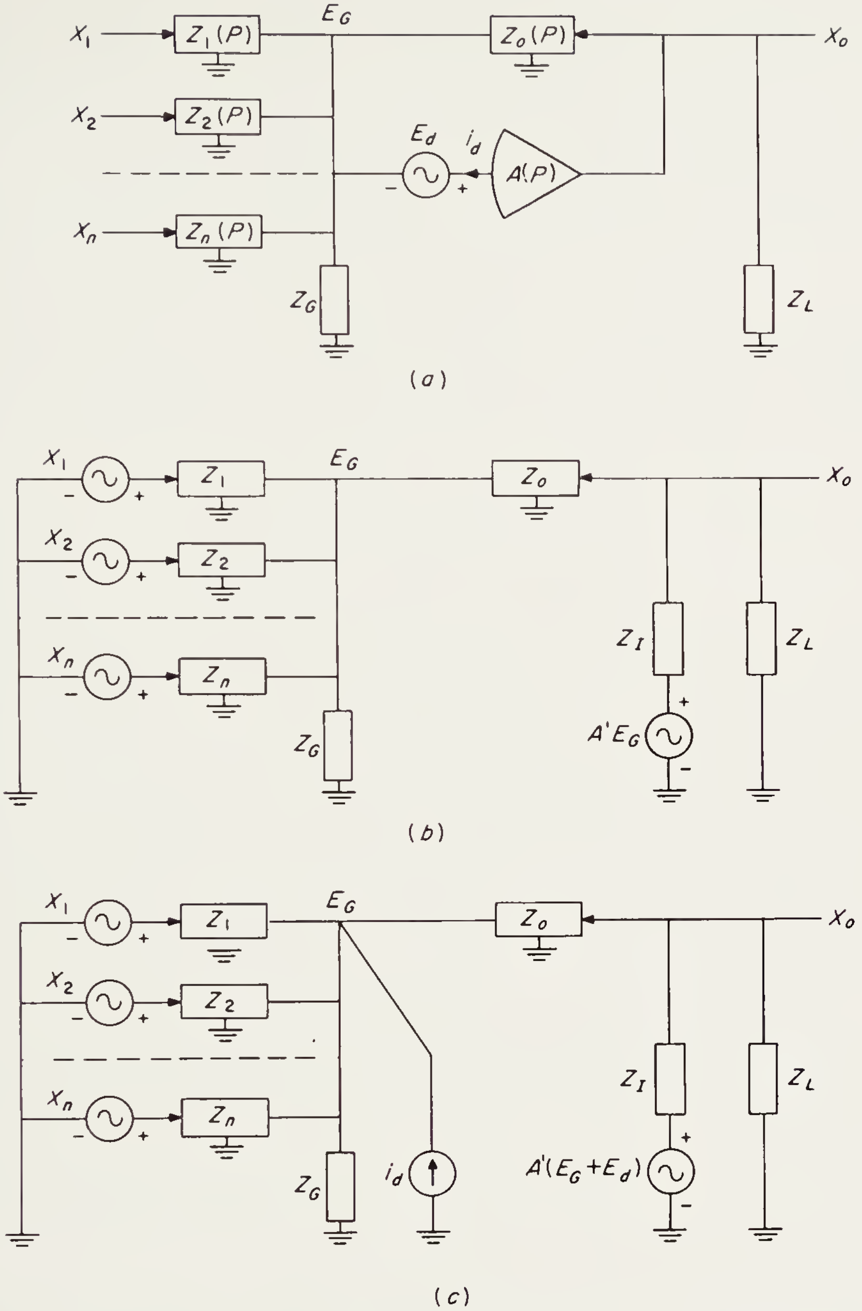


Fig. 3-15. Operational amplifier (*a*) and equivalent circuits (*b*) and (*c*).

will be simply added to the output due to the given input voltages $X_1, X_2, \ldots, X_n$, we can leave a discussion of these effects safely for Sec. 3-15 and start by neglecting the offset/noise voltage E_d and current i_d. This results in the simplified equivalent circuit of Fig. 3-15*b*, while Fig. 3-15*c* includes sources introducing E_d and i_d. Referring to Fig. 3-15*b*, elimination of all node voltages except for X_1, $X_2, \ldots, X_n$, E_G, and X_0 will yield the node equations for the summing point and for the output terminal in the respective forms

$$E_G = \alpha_1(P)X_1 + \alpha_2(P)X_2 + \cdots + \alpha_n(P)X_n + \beta(P)X_0 \qquad (3\text{-}16)$$

and

$$X_o = a_I(P)A'(P)E_G + a_T(P)E_G = A(P)E_G \qquad (3\text{-}17a)$$

where $A'(P)$ is the open-loop/open-circuit gain of the amplifier used, and

$$A = A'\left(a_I + \frac{a_T}{A'}\right) \qquad (3\text{-}17b)$$

Combination of Eqs. (16) and (17) yields the exact performance equation

$$X_0 = \frac{A\beta}{1 - A\beta}\,\frac{\alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_n X_n}{\beta} \qquad (3\text{-}18)$$

or, since Eqs. (15) and (18) must agree as $|A'(j\omega)| \to \infty$ with $a_I\beta \neq 0$,

$$\begin{aligned} X_0 &= \frac{A\beta}{1 - A\beta} Z_0\left(\frac{X_1}{Z_1} + \frac{X_2}{Z_2} + \cdots + \frac{X_n}{Z_n}\right) \\ &= -\left(1 - \frac{1}{1 - A\beta}\right) Z_0\left(\frac{X_1}{Z_1} + \frac{X_2}{Z_2} + \cdots + \frac{X_n}{Z_n}\right) \end{aligned} \qquad (3\text{-}19)$$

In practice, the transfer impedances $Z_0, Z_1, Z_2, \ldots, Z_n$ can be found in published tables (Table A-1). The transfer operators β, a_I, and a_T must be obtained for each specific operational amplifier from the explicit node equations (16), (17). β, a_I, and a_T admit an intuitively suggestive interpretation:

β is the *feedback ratio* (output terminal to summing point), so that $A\beta$ is the *loop gain* of the operational amplifier.

a_I is the attenuation of the open-circuit forward gain A' due to amplifier loading by computing networks and load ($a_I = 1$ if $Z_I = 0$).

a_T is the "feedforward" transfer function from summing point to output; a_T/A' measures an undesirable lack of isolation between inputs and output.

In certain transistor amplifiers, the output voltage X_0 is proportional to the current into Z_G, and Z_G is small compared with the computing impedances. In this case, $A' = K/Z_G$, $\beta \approx bZ_G$, and $A'\beta \approx Kb$ is practically independent of Z_0 (current amplifier).

3-12. Error Calculations. The exact performance equation (19) permits convenient separation and estimation of the error or difference

$$e = \frac{1}{1 - A\beta} Z_0\left(\frac{X_1}{Z_1} + \frac{X_2}{Z_2} + \cdots + \frac{X_n}{Z_n}\right) \qquad (3\text{-}20)$$

between the actual amplifier output (19) and the output predicted by the approximate performance equation (15), which is commonly used to design the amplifier circuit. Substitution of $j\omega$ for P in Eq. (19) yields the *phase error* $\delta(\omega)$ for steady-state sinusoidal inputs at the circular frequency ω:

$$\delta(\omega) = \arg\frac{-A\beta}{1 - A\beta} = -\arg\left(1 - \frac{1}{A\beta}\right) \quad \text{radians} \tag{3-21}$$

for the usual large values of $|A| = |A(j\omega)|$ and $|A\beta| = |A(j\omega)\beta(j\omega)|$, one has

$$e \approx -\frac{1}{A\beta} Z_0\left(\frac{X_1}{Z_1} + \frac{X_2}{Z_2} + \cdots + \frac{X_n}{Z_n}\right) \tag{3-22}$$

and the absolute percentage dynamic error $|\epsilon(\omega)|$ for steady-state sinusoidal inputs may be estimated from

$$|\epsilon(\omega)| \approx 100\frac{1}{|A(j\omega)\beta(j\omega)|} \tag{3-23}$$

Equations (19) to (23) specify the error arising from the use of the simplified design equation (15) in the face of finite gain and amplifier loading. Equation (20) or (23) may be used to define the working frequency range of the operational amplifier for a given percentage error $|\epsilon(\omega)|$. Near the limits of the working frequency range, $|\epsilon(\omega)|$ is often numerically practically equal to $100|\delta(\omega)|$; i.e., most of the dynamic error is due to phase shift (see also Secs. 3-1, 3-6, 3-19, and 3-20).

Quite frequently, $|a_T/A'|$ can be neglected compared with a_I, so that the steady-state absolute fractional dynamic error $100|\epsilon(\omega)|$ is simply the reciprocal of the absolute loop gain $|A'a_I\beta|$ and can be studied very conveniently by means of the open-loop Bode plot for the operational amplifier (see also Secs. 3-19 and 3-20).

3-13. Effect of Amplifier Internal Impedance and Loading. The equivalent-generator impedance (source impedance) $Z_S(P)$ of the operational amplifier, as seen by the load, is given by

$$X_0 = \frac{Z_L}{Z_S + Z_L}\bar{X}_0 \tag{3-24}$$

where X_0 is given by Eq. (19), and $\bar{X}_0$ is the same expression with $Z_L = \infty$ (no-load output voltage). Equations (19) and (24) yield

$$Z_S = \left(\frac{\bar{X}_0}{X_0} - 1\right)Z_L = \left(\frac{1 - 1/A\beta}{1 - 1/\bar{A}\beta} - 1\right)Z_L \tag{3-25}$$

where barred terms refer to the no-load condition ($Z_L = \infty$). Introducing Eq. (17*b*) and neglecting terms of higher order in $1/A'$, one finds

$$Z_S \approx \frac{1}{A'\beta}\left(\frac{1}{\bar{a}_I} - \frac{1}{a_I}\right)Z_L$$

where $$a_I = \frac{Z_L Z_F}{Z_L Z_F + Z_L Z_I + Z_I Z_F} \qquad \bar{a}_I = \frac{Z_F}{Z_I + Z_F}$$

if Z_F is the load due to the feedback network. It follows that

$$Z_S \approx -\frac{Z_I}{A'\beta} \tag{3-26}$$

3-14. Operational Amplifiers with Two-terminal Computing Impedances. Input-source Loading. (a) In the important special case of operational amplifiers with simple two-terminal computing impedances $Z_0, Z_1, Z_2, \ldots, Z_n$ (Fig. 1-15*b*), the transfer impedances are ordinary impedances. The summing-point node equation (16) becomes

$$\frac{1}{Z_0}(E_G - X_0) + \frac{1}{Z_1}(E_G - X_1) + \frac{1}{Z_2}(E_G - X_1) + \cdots$$
$$+ \frac{1}{Z_n}(E_G - X_n) + \frac{1}{Z_G} E_G = 0 \tag{3-27}$$

and the output-terminal node equation (17) is now

$$\frac{1}{Z_0}(X_0 - E_G) + \frac{1}{Z_I}(X_0 - A'E_G) + \frac{X_0}{Z_L} = 0 \tag{3-28}$$

By comparison of Eqs. (16), (17), and (27), (28), it follows that

$$\left.\begin{aligned}
\beta &= \frac{1}{Z_0}\left(\frac{1}{Z_0} + \frac{1}{Z_1} + \frac{1}{Z_2} + \cdots + \frac{1}{Z_n} + \frac{1}{Z_G}\right)^{-1} = \frac{Z_A}{Z_0} \\
a_I &= \frac{Z_0 Z_L}{Z_I Z_0 + Z_I Z_L + Z_0 Z_L} = \frac{Z_B}{Z_I} \\
a_T &= \frac{Z_I Z_L}{Z_I Z_0 + Z_I Z_L + Z_0 Z_L} = \frac{Z_B}{Z_0}
\end{aligned}\right\} \tag{3-29}$$

$$Z_S = \frac{Z_0 Z_I}{(1 - A'\beta)Z_0 + (1 - \beta)Z_I}\left[\approx -\frac{Z_I}{A'\beta} \text{ if } |A'(j\omega)\beta(j\omega)| \gg 1\right] \tag{3-30}$$

where Z_A is the combined parallel impedance of all branches terminating at the summing point ($Z_0, Z_1, Z_2, \ldots, Z_n$, and Z_G), and Z_B is the combined parallel impedance of the branches terminating at the output terminal (Z_0, Z_I, and Z_L). The expressions (29) and (30) must be substituted into the design equations (19) to (26).

(b) Input-source Loading. If Z_1 (but not necessarily $Z_0, Z_2, Z_3, \ldots, Z_n$) is a two-terminal impedance, then the source of the input voltage X_1 sees the impedance Z_1 grounded in series with the voltage

$E_G = X_0/A$, or an equivalent input impedance

$$Z_1' = -\frac{Z_1}{1 - \dfrac{\beta}{1 - A\beta}\dfrac{Z_0}{Z_1}} \tag{3-31}$$

grounded in series with the voltage

$$\frac{\beta}{1 - A\beta} Z_0 \left(\frac{X_2}{Z_2} + \frac{X_3}{Z_3} + \cdots + \frac{X_n}{Z_n}\right)$$

For sinusoidal components such that $|A(j\omega)\beta(j\omega)| \gg 1$, the X_1 source sees essentially the grounded impedance Z_1; the summing point of a high-gain parallel-feedback operational amplifier acts as a "virtual ground" which effectively isolates each input from all others and from the amplifier output.

3-15. Effects of D-c Offsets and Noise. In Fig. 3-15*a*, the voltage E_d represents the combined effect of *d-c unbalance* due to initial unbalance or drift and/or *noise* due to vacuum tubes, transistors, resistors, choppers, and stray pickup, all referred to the amplifier input. The current i_d represents similar effects described in terms of currents into the amplifier summing point; i_d is mainly direct offset current due to grid or base current in the amplifier input stage, and leakage from various terminals into the summing point. The latter effect can be eliminated by grounded conductors surrounding all summing-point terminations (see also Sec. 7-4*a*).

The effects of E_d and i_d must be added to the output (19) computed in Sec. 3-11. Referring to the equivalent circuit of Fig. 3-15*c*, we repeat the analysis of Sec. 3-11 to find that Eq. (19) for the amplifier output voltage is now replaced by

$$X_0 = -\left(1 - \frac{1}{1 - A\beta}\right) Z_0 \left(\frac{X_1}{Z_1} + \frac{X_2}{Z_2} + \cdots + \frac{X_n}{Z_n}\right) + \frac{a_I A'}{1 - A\beta} E_d + \frac{A\beta}{1 - A\beta} Z_0 i_d \tag{3-32}$$

By virtue of the superposition principle, the absolute error due to E_d and i_d, viz.,

$$e_d = \frac{a_I A'}{1 - A\beta} E_d + \frac{A\beta}{1 - A\beta} Z_0 i_d \tag{3-33}$$

is simply added to the error (20) due to finite gain and loading. For d-c offset, and for low-frequency noise, one has usually $|A(j\omega)\beta(j\omega)| \gg 1$, $a_I A' \approx A$, so that

$$e_d \approx -\frac{1}{\beta} E_d - Z_0 i_d \tag{3-34}$$

3-16. Effects of Gain Changes and Distortion. The inverse feedback inherent in the operational-amplifier circuit helps to reduce the effects of gain changes in the d-c amplifier on the output voltage X_0; the range of permissible gain changes may be investigated with the aid of Eqs. (19) and (22).

For pure sinusoidal voltages, the fractional effect of a steady-state gain change $dA(j\omega)$ on the output amplitude is reduced in the ratio $1:|1 - A(j\omega)\beta(j\omega)|$. A similar reduction applies to the fractional amplitude of each separate sinusoidal distortion and intermodulation component originating in the d-c amplifier, but the effect of feedback on the *total* distortion error depends on the vector sum of the distortion components and is not so easily computed.

3-17. Summary of Error Analysis. The analysis of Secs. 3-10 to 3-16 indicates that, with ideal network components, all operational-amplifier errors other than d-c offset are, for all practical purposes, inversely proportional to the absolute feedback loop gain $|A(j\omega)\beta(j\omega)|$. With modern high-gain low-offset d-c amplifiers, computing accuracy is therefore limited essentially by calibration errors and by parasitic-impedance effects, which will be further investigated in Secs. 3-18 to 3-20.

An additional consideration is the accumulation of errors due to phase shift, distortion, d-c offset, and noise in cascaded strings or loops of operational amplifiers. As a rough general rule, one obtains the best results by keeping the absolute loop gains $|A(j\omega)\beta(j\omega)|$ (and hence the closed-loop forward gains, assuming similar d-c amplifiers) of the various amplifiers approximately equal near the most typical working frequencies.

DESIGN OF PHASE INVERTERS, SUMMING AMPLIFIERS, AND INTEGRATORS

3-18. Introduction. The design of specific operational amplifiers such as summers or integrators requires us to combine the design criteria of Secs. 3-10 to 3-17 with due consideration of parasitic impedances in physical networks (Secs. 3-7 to 3-9). The designer is, furthermore, constrained by system considerations, such as amplifier power and potentiometer loading (Sec. 3-24).

3-19. Design of Phase Inverters and Summing Amplifiers. (a) Static Accuracy. For a phase inverter or summing amplifier (Fig. 3-16), the d-c feedback ratio is

$$\beta(0) = \left(\frac{R_0}{R_1} + \frac{R_0}{R_2} + \cdots + \frac{R_0}{R_n} + \frac{R_0}{R_G} + 1\right)^{-1} \le \frac{1}{a+1} \qquad (3\text{-}35)$$

where a is simply the absolute phase-inverter gain, or the total absolute summer gain (phase-inverter gain with all summer inputs connected together) at d-c. Referring to Eq. (20), we see that *absolute static accuracy within ϵ percent requires a d-c gain $|A(0)|$ exceeding*

$$|A(0)|_{\text{MIN}} = (a+1)\left(\frac{100}{\epsilon} - 1\right) \approx \frac{100}{\epsilon}(a+1) \qquad \text{(3-36)}$$

in addition to correspondingly accurate matching of the resistance ratios R_0/R_i and low amplifier drift. Modern chopper-stabilized d-c amplifiers exceed the low-frequency gain requirement (36) by a wide margin and have very low drift (Sec. 3-36). The static accuracy of phase inverters and summers in general-purpose analog computers is, then, essentially determined by resistance-ratio accuracy.

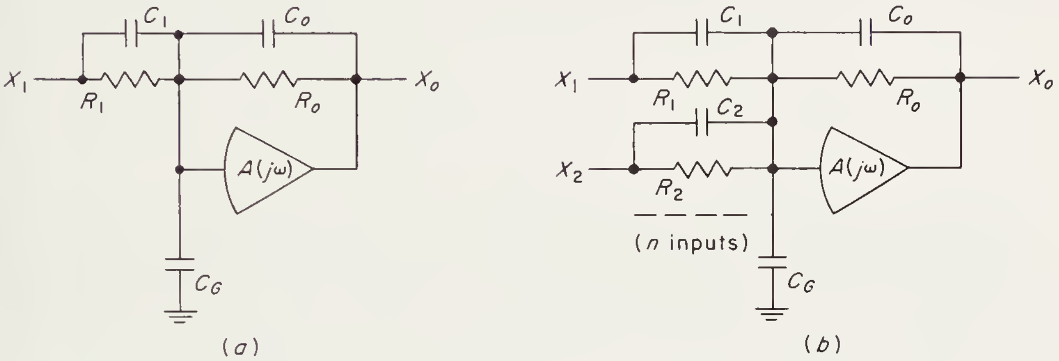


Fig. 3-16. Phase-inverting amplifier (a) and summing amplifier (b).

EXAMPLE: For absolute static accuracy within 0.01 percent, a summing amplifier with 1, 1, 1, 10, 10, 10 inputs ($a = 33$) requires a d-c gain $|A(0)| \geq 34 \times 10^4$. $|A(0)| \geq 10^7$ and d-c offset below 100 μV (i.e., 3.4 mV at the amplifier output, Sec. 3-15) are easy to obtain.

(b) Dynamic Accuracy and Phase-equalizing Capacitors. Figure 3-17a shows the open-loop frequency response of a typical computer d-c amplifier (see also Sec. 3-31). *Although the newer amplifiers maintain open-loop gains $|A(j\omega)|$ above* 1,000 *well beyond* 10 *KHz* (Secs. 3-31 and 3-35), *phase shift due to the parasitic capacitances C_G, C_0, C_1, C_2, . . . , C_n indicated in Fig.* 3-16 *affects phase-inverter or summer accuracy at much lower frequencies.* In the following, we can restrict explicit analysis to the case of the phase inverter, since the summing-amplifier performance will be, essentially, that of a phase inverter with the input resistance

$$\left(\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}\right)^{-1}$$

paralleled by the capacitance $C_1 + C_2 + \cdots + C_n$.

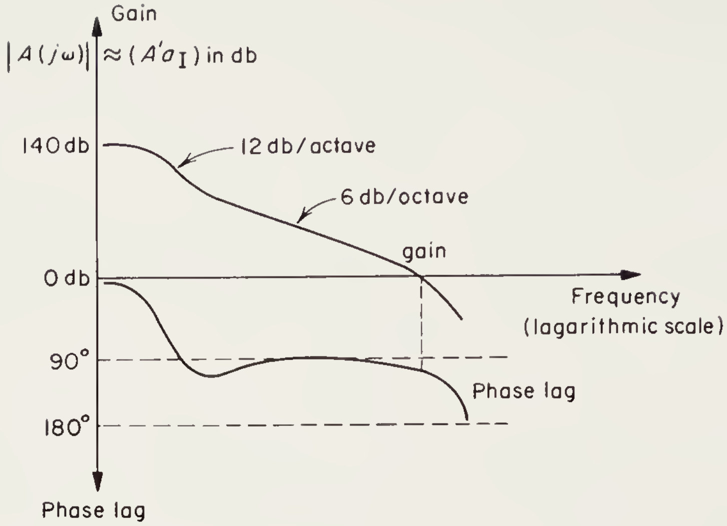


Fig. 3-17*a*. Open-loop frequency response of a computer d-c amplifier. In the low-frequency region, stabilizer-channel filters and rolloff networks shape a more or less conservative amplitude and phase response. In the intermediate-frequency region, rolloff networks and/or feedforward circuits reduce slope and phase shift. Circuit and load capacitances cause the response to drop sharply after unity gain (0 db) is reached (see also Sec. 3-34).

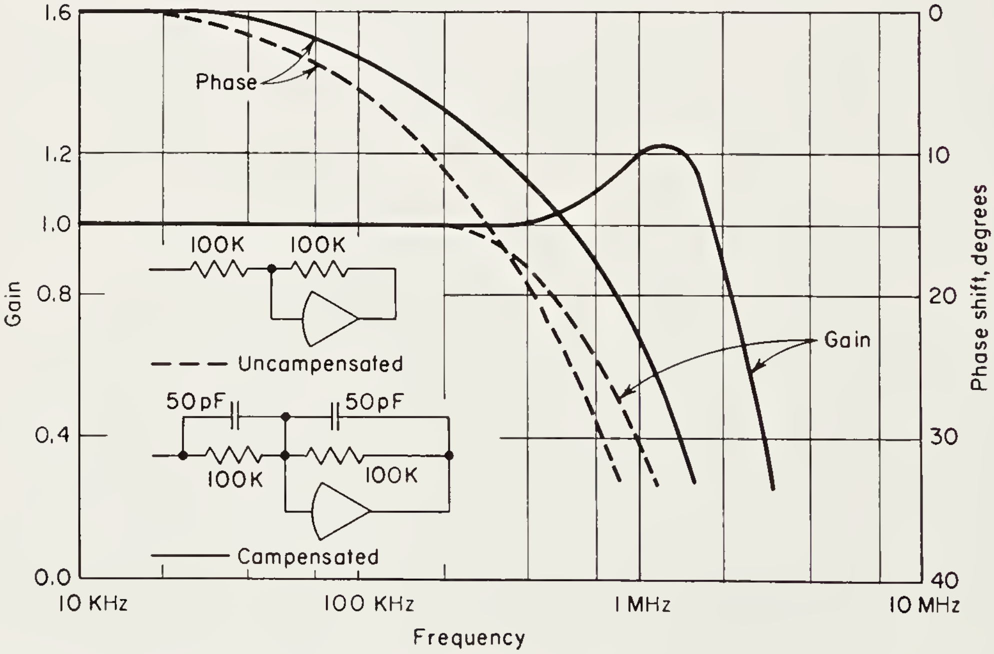


Fig. 3-17*b*. Unity-gain phase-inverter response with and without phase-compensating capacitors. (*University of Arizona.*)

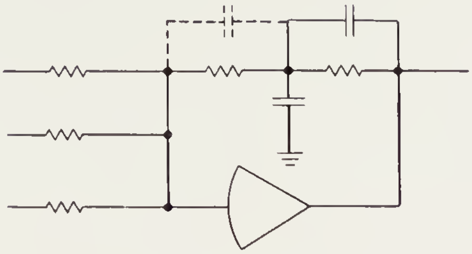


Fig. 3-17*c*. Modern fast summers and phase inverters substitute T-network equalization for the shunt capacitors in Fig. 3-17*b*. The T circuit is cheaper, easier to adjust, and eliminates the extra capacitive load on preceding amplifiers. Capacitors are usually selected by trial and error. (*University of Arizona*, *Ref*. 65.)

Referring to Fig. 3-16, we write the amplifier open-loop gain as

$$A(j\omega) = -\frac{A_0}{1 + j\delta_A(\omega)} \tag{3-37}$$

where A_0 is the absolute d-c gain; $\delta_A(\omega)$ is, in general, complex, but most frequently real and approximately proportional to ω over a wide frequency range (Fig. 3-17*a*). The shunt capacitance C_G is due to wiring and interelectrode capacitances, including Miller effect (Refs. 34 and 36). C_1 and C_0 are resistor capacitances (Sec. 3-7) combined with added equalizing capacitances.

Figure 3-16 yields the phase-inverter (or summer) frequency response

$$\frac{\vec{X}_0}{\vec{X}_1} = -\frac{R_0}{R_1}\frac{j\omega R_1 C_1 + 1}{j\omega R_0 C_0 + 1}\left[1 - \frac{1}{1 - A(j\omega)\beta(j\omega)}\right] \tag{3-38a}$$

$$\text{with}\quad \beta(j) = \frac{1}{R_0/R_1 + 1}\,\frac{j\omega R_0 C_0 + 1}{j\omega\dfrac{R_0}{1 + R_0/R_1}(C_0 + C_1 + C_G) + 1} \tag{3-38b}$$

either directly or by reference to Sec. 3-14. The desired response is $-R_0/R_1$. We expand Eq. (38) with the aid of the relation

$$\frac{1+\lambda}{1+\mu} = (1+\lambda)(1 - \mu + \mu^2 \mp \cdots) \qquad (|\mu| < 1) \tag{3-39}$$

At frequencies sufficiently low for us to neglect higher-order terms,

$$\begin{aligned}\frac{\vec{X}_0}{\vec{X}_1} \approx -\frac{R_0}{R_1}\Bigg\{1 &+ \omega^2 R_0 C_0(R_1 C_1 - R_0 C_0) - \frac{1}{A_0}\left(\frac{R_0}{R_1} + 1\right) \\ &+ j\omega\Bigg[R_1 C_1 - R_0 C_0 + \frac{R_0}{A_0 R_1}(R_0 C_0 - R_1 C_1 - R_1 C_G) \\ &\qquad - \left(\frac{R_0}{R_1} + 1\right)\frac{\delta_A(\omega)}{\omega A_0}\Bigg]\Bigg\}\end{aligned} \tag{3-40}$$

This analysis yields important and practical design information:

1. Dynamic as well as static errors increase with the closed-loop gain $a = R_0/R_1$. This applies, in particular, to the output amplifiers of electronic function generators and multipliers.
2. *We can often extend phase-inverter or summer frequency response by at least a decade if we pad the resistor shunt capacitances C_0, C_1 so as to cancel error terms in Eq.* (40) (Fig. 3-17).

We shall try to make R_1C_1 slightly larger than R_0C_0 to partially compensate the effects of C_G and $\delta_A(\omega)$. $R_1C_1 - R_0C_0$ is reduced to the order of magnitude of $1/A_0$, so that Eq. (40) is replaced by

$$\frac{\vec{X}_0}{\vec{X}_1} \approx -\frac{R_0}{R_1}\left\{1 - \frac{1}{A_0}\left(\frac{R_0}{R_1} + 1\right)\right. \\ \left. + j\omega\left[R_1C_1 - R_0C_0 - \frac{R_0C_G}{A_0} - \left(\frac{R_0}{R_1} + 1\right)\frac{\delta_A(\omega)}{\omega A_0}\right]\right\} \qquad (3\text{-}41a)$$

The corresponding phase error is

$$\delta_S(\omega) = \arg\left(-\frac{\vec{X}_0}{\vec{X}_1}\right) \approx \omega\left(R_1C_1 - R_0C_0 - \frac{R_0C_G}{A_0}\right) \\ - \frac{1}{A_0}\left(\frac{R_0}{R_1} + 1\right)\delta_A(\omega) \qquad (3\text{-}41b)$$

where $\delta_S > 0$ signifies phase *lead.*

Padding of C_0, C_1 can benefit older existing computers as well as new designs.[13,14] In general, both C_0 and C_1 must be increased; note that the resulting pole-zero cancellation in the feedback ratio (38*b*) also benefits amplifier *stability* (Sec. 3-34) and may thus permit increased gain. Single[13,14] recommends about 40 pF shunt capacitance for 1M resistors in "slow" analog computers. The corresponding 400 pF shunt capacitance for 100K resistors is already a rather uncomfortable load on preceding amplifiers and especially on coefficient-setting potentiometers (Sec. 3-6). In wideband computing amplifiers operating above 1 KHz, frequency response depends so critically on shunt-capacitance adjustments that individual trimmer capacitors and/or (preferably) lower computing-resistance values are a practical necessity. Resistances in series with C_1 and C_0 can be useful for "stopping" the capacitive loading at high frequencies. Such frequency-dependent loads, however, cannot usually be connected to coefficient-setting potentiometers.

3-20. Electronic Integrators: Effects of Finite Gain, Loading, and Parasitic Impedances on the Integrator Frequency Response. Figure 3-18*a* shows an electronic-integrator model without many of the

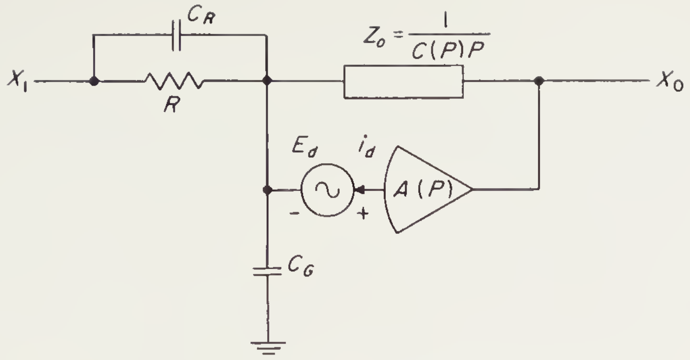


Fig. 3-18a. Electronic integrator with finite gain, loading, parasitic impedances, and d-c offset.

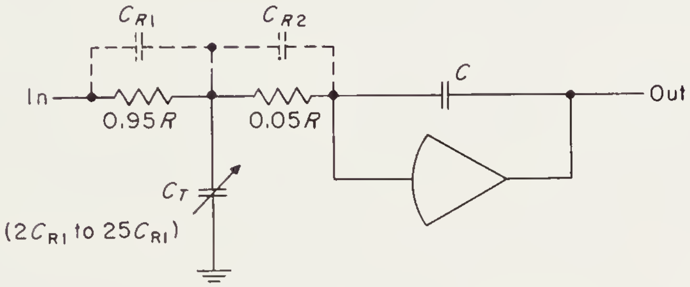


Fig. 3-18*b*. A simple way to reduce integrator phase error due to the parasitic capacitance C_R is to use a series combination of two or three metal-film resistors for R. For more precise phase compensation, a trimmer C_T is added; C_{R2} is of the same order of magnitude as C_{R1} (typically 0.5 to 2 pF), so that $0.05RC_{R2} \ll RC_{R1}$.

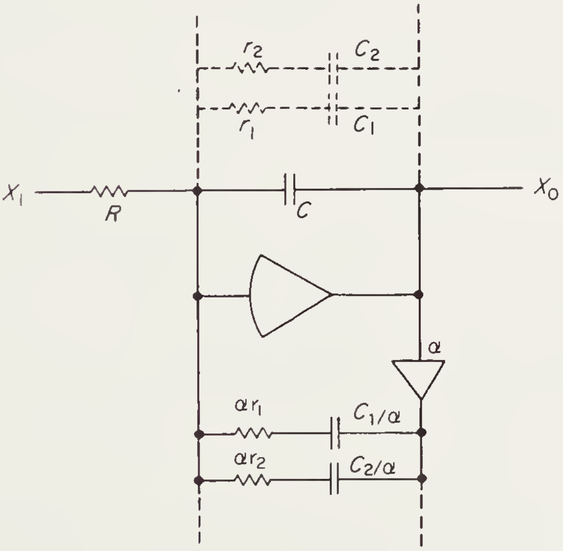


Fig. 3-18*c*. Compensation of capacitor dielectric absorption in an accurate integrator. *RC* feedback circuits with time constants equal to those of the Dow capacitor model (Fig. 3-13*a*) are driven by an inexpensive inverter. Attenuation in the inverter circuit permits one to use more convenient resistance and capacitance values. The effective dielectric absorption can be reduced by a factor of at least 10. (*C. H. Single.*[23])

idealizing assumptions made in Sec. 1-2:

1. A suitably given expression for the amplifier gain $A = A(P)$ will account for effects of *finite gain* and can be written in the form (37) to account for the effect of the *amplifier load.*
2. The "capacitance" $C = C(P)$ is not a constant, but is given by the capacitor model of Sec. 3-9 as

$$C(P) = C(\infty) + \frac{1}{r_{DC}P} + \sum_{k=1}^{n} \frac{C_k}{1 + r_k C_k P} \tag{3-42a}$$

$$\text{or} \quad C(j\omega) = C_{\text{EFF}}(\omega) + \frac{1}{j\omega R_D(\omega)} = C_{\text{EFF}}(\omega)[1 - jD(\omega)] \tag{3-42b}$$

   to account for both *resistive leakage* and *dielectric absorption.*
3. The *distributed capacitance* C_R *of the integrating resistor* and the *summing-point capacitance* C_G are explicitly included. Resistor inductance can safely be neglected below 200 KHz (Sec. 3-7).

The purely additive effects of d-c offset and noise (E_d and i_d in Fig. 3-18*a*) will again be considered separately (Sec. 3-21). And, as in Sec. 3-19, our frequency-response analysis also applies to summing integrators (Sec. 1-10) if R is regarded as the parallel combination of all summing resistors, and C_R is the sum of their shunt capacitances.

For $E_d = i_d = 0$, Fig. 3-18*a* yields the summing-point node equation

$$\left(\frac{1}{R} + C_R P\right)\left[\frac{1}{A(P)} X_0 - X_1\right] + \frac{C_G P}{A(P)} X_0 + C(P)P\left[\frac{1}{A(P)} X_0 - X_0\right] = 0 \tag{3-43}$$

and hence the *integrator performance equation*

$$X_0 = -\frac{RC_R P + 1}{RC(P)P - [RP/A(P)][C(P) + C_G + C_R] - [1/A(P)]} X_1 \tag{3-44a}$$

We can substitute the proper expressions for $C(P)$ and $A(P)$ into this equation to compare our integrator performance with ideal integration ($X_0 = \text{const}/P$).

For steady-state sinusoidal signals, we introduce

$$C(j\omega) = C_{\text{EFF}}(\omega)[1 - jD(\omega)]$$

where $C_{\text{EFF}}(\omega)$ is the "effective" or "nominal" integrating capacitance measured with sinusoidal signals at the circular frequency ω. Substitution of $j\omega$ for P in Eq. (44*a*) yields the *integrator frequency-response function*

$$\frac{\vec{X}_0}{\vec{X}_1} = -\frac{1}{j\omega RC_{\text{EFF}}} \cdot \frac{j\omega RC_R + 1}{1 - jD(\omega) - \dfrac{1}{A(j\omega)}\left[1 - jD(\omega) + \dfrac{C_G + C_R}{C_{\text{EFF}}} + \dfrac{1}{j\omega RC_{\text{EFF}}}\right]} \tag{3-44b}$$

which must be compared with the frequency response $-1/j\omega RC_{\text{EFF}}$ of an ideal integrator whose capacitance equals the "effective" capacitance $C_{\text{EFF}} = C_{\text{EFF}}(\omega)$ at the frequency in question. At this point, it is again expedient to write the amplifier again $A(j\omega)$ in the form (37). For reasonably small errors, we can now expand the frequency-response function (44*b*) with the aid of the relation (39) and find

$$\begin{aligned}\frac{\vec{X}_0}{\vec{X}_1} \approx -\frac{1}{j\omega RC_{\text{EFF}}}\biggl\{1 + j\biggl[\frac{1}{\omega A_0 RC_{\text{EFF}}} + D(\omega) + \omega RC_R \\ - \frac{\delta_A(\omega)}{A_0}\left(1 + \frac{C_G}{C_{\text{EFF}}}\right)\biggr] - \frac{1}{A_0}\biggl[1 + \frac{C_G + C_R}{C_{\text{EFF}}} + D(\omega)\delta_A(\omega) \\ + \frac{\delta_A(\omega)}{\omega RC_{\text{EFF}}} - jD(\omega) + j\delta_A(\omega)\frac{C_R}{C_{\text{EFF}}}\biggr]\biggr\}\end{aligned} \tag{3-45}$$

The terms obtained from the last bracket can be conservatively neglected compared with the other error terms in Eq. (45) for essentially every practical integrator circuit; this fact should be checked in each particular case (Sec. 3-24). *The frequency response of a practical electronic integrator is, then, given by*

$$\begin{aligned}\frac{\vec{X}_0}{\vec{X}_1} \approx -\frac{1}{j\omega RC_{\text{EFF}}}\biggl\{1 + j\biggl[\frac{1}{\omega A_0 RC_{\text{EFF}}} + D(\omega) \\ + \omega RC_R - \frac{\delta_A(\omega)}{A_0}\left(1 + \frac{C_C}{C_{\text{EFF}}}\right)\biggr]\biggr\}\end{aligned} \tag{3-46}$$

throughout the integrator working frequency range. The resulting *integrator phase error* is

$$\begin{aligned}\delta_I(\omega) = \arg\frac{\vec{X}_0}{\vec{X}_1} - \frac{\pi}{2} \approx \frac{1}{\omega A_0 RC_{\text{EFF}}} + D(\omega) \\ + \omega RC_R - \frac{\delta_A(\omega)}{A_0}\left(1 + \frac{C_G}{C_{\text{EFF}}}\right)\end{aligned} \tag{3-47}$$

(phase *lead* for $\delta_I > 0$), and the absolute percentage error for sinusoidal components at the circular frequency ω is $|\varepsilon_I(\omega)| = 100|\delta_I(\omega)|$ (see also Sec. 3-2). The error in Eq. (46) is seen to be a sum of terms respectively due to *finite amplifier gain* (at d-c or at the computer repetition rate), *capacitor leakage and dielectric absorption, distributed capacitance of the integrating resistor*, and *amplifier high-frequency bandwidth limitations.*

In practice, C_G/C_{EFF} is small for all but very fast integrators. The capacitor dissipation coefficient $D(\omega)$ is usually almost constant except at very low frequencies, where it is given by $1/\omega r_{DC}C(\infty)$ (Sec. 3-9). The phase shift due to C_R precludes the use of 1M integrating resistors above about 50 Hz. Figure 3-18*b* and *c* show possible compensation schemes.[96] Table 3-2 illustrates the relative importance of the various error sources with two numerical examples.

3-21. Electronic Integrators: Errors Due to Calibration, D-c Offset, and Noise. In addition to the dynamic errors discussed in Sec. 3-20, integrators exhibit a constant-percentage *calibration error* mainly due to changes in the effective capacitance C_{EFF} with temperature (and also with frequency, Secs. 3-8 and 3-9). Finally, *d-c offset voltage and/or current* (Fig. 3-18*a*) produce an additive error output

$$X_o = -\frac{\tau}{RC}(E_d + Ri_d) \tag{3-48}$$

Table 3-2. Examples of Error Sources in Electronic Integration

(Secs. 3-20 and 3-21.) For comparison purposes note that *unity-gain phase-inverter errors* are usually readily made less than integrator errors (Sec. 3-19), while *errors due to potentiometer phase shift* might exceed the figures quoted for $100\omega RC_R$ percent by a factor of 10 without careful phase compensation (Sec. 3-6).

Source	A typical "slow" integrator		A low-cost repetitive-computer integrator		A high-quality wideband integrator	
Half-scale	10 or 100 volts		10 or 100 volts		10 volts	
Calibration error	0.025 percent		0.5 percent		0.3 percent	
C, R	1 μF, 1M		0.001 μF, 50K		0.005 μF, 2K	
$A_0, \delta_A(\omega)$	10^7, 10ω		10^4, $\omega/100$		10^6, $\omega/100$	
D	2×10^{-4}		$<10^{-3}$		$<10^{-3}$	
C_R	4 pF		2 pF		1 pF	
C_G	500 pF		20 to 100 pF		20 to 50 pF	
E_d, i_d	50 μV		200 μV, 20 nA*		100 μV, 10 nA*	
Frequency	0.01 Hz†	100 Hz	100 Hz†	10 KHz	1 KHz†	100 KHz
$\frac{100}{\omega A_0 RC}$ (percent)	1.6×10^{-4}	1.6×10^{-8}	3.2×10^{-2}	3.2×10^{-4}	1.6×10^{-3}	1.6×10^{-5}
$-100\frac{\delta_A(\omega)}{A_0}$ (percent)	-6.3×10^{-6}	-6.3×10^{-2}	-6.3×10^{-2}	-6.3	-6.3×10^{-3}	-0.63
$100D$ (percent)	2×10^{-2}	2×10^{-2}	<0.1	<0.1	<0.1	<0.1
$100\omega RC_R$ (percent)	2.5×10^{-5}	0.25	6.3×10^{-2}	0.63	1.25×10^{-3}	0.125
$\frac{E_d}{RC}\tau$ (per cycle)	5 mV	0.5 μV	4 mV	40 μV	10 mV	100 μV
$\frac{i_d}{C}\tau$ (per cycle)	0	0	200 mV	2 mV	2 mV	20 μV

* From electronic switch and/or transistor amplifier.

† One cycle at this frequency is one typical computer run with the R, C values quoted.

where τ is the computing time. Effects of high-frequency noise are largely "integrated out."

For modern high-quality d-c amplifiers, E_d will be between 10 and 100 μV, while i_d is of the order of nanoamperes even in the case of transistor amplifiers (Sec. 3-33). The integrator drift error (48) is, then, usually appreciable only in instrumentation applications requiring long integration times; in general-purpose analog computers, integrator accuracy is most frequently limited by capacitance calibration and dielectric absorption (see also Table 3-2).

ERRORS IN SOLUTIONS OF LINEAR DIFFERENTIAL EQUATIONS WITH CONSTANT COEFFICIENTS. SYSTEM-DESIGN CONSIDERATIONS

3-22. Introduction. Consider a set of potentiometers, summer/phase inverters, integrators, and/or more general operational amplifiers interconnected to solve a linear homogeneous differential equation with constant coefficients,

$$F(P)Y \equiv (a_nP^n + a_{n-1}P^{n+1} + \cdots + a_0)Y = 0 \qquad \left(P \equiv \frac{d}{d\tau}\right) \qquad (3\text{-}49)$$

To assess the effects of errors in the various computing elements, we replace their idealized performance equations by the corresponding more realistic relations from Secs. 3-6, 3-11, 3-19, and 3-20. The analog-computer setup will, then, implement a linear differential equation

$$\bar{F}(P)\bar{Y} \equiv (\bar{a}_{\bar{n}}P^{\bar{n}} + \bar{a}_{\bar{n}-1}P^{\bar{n}-1} + \cdots + \bar{a}_0)\bar{Y} = 0 \qquad \left(P \equiv \frac{d}{d\tau}\right) \qquad (3\text{-}50)$$

different from the desired given differential equation (49) and, in general, of higher order than the given equation ($\bar{n} \geq n$). In particular, the n roots $s_1, s_2, \ldots, s_n$ of the characteristic equation

$$F(s) \equiv a_ns^n + a_{n-1}s^{n-1} + \cdots + a_0 = 0 \qquad (3\text{-}51)$$

which determine the normal modes of the correct solution $Y = Y(\tau)$ are replaced by the $\bar{n}$ roots $\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_{\bar{n}}$ of the new characteristic equation

$$\bar{F}(s) \equiv \bar{a}_{\bar{n}}s^{\bar{n}} + \bar{a}_{\bar{n}-1}s^{\bar{n}-1} + \cdots + \bar{a}_0 = 0 \qquad (3\text{-}52)$$

associated with the actual computer setup. Small phase and amplitude errors in the various linear computing elements tend to produce a twofold effect:

1. n of $\bar{n}$ roots of Eq. (52), say $\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_n$, will correspond to the n correct roots $s_1, s_2, \ldots, s_n$, but will be slightly shifted; this

expresses errors in the corresponding normal modes of the computer solution $\bar{Y} = \bar{Y}(\tau)$.

2. The $\bar{n} - n$ remaining roots $\bar{s}_{n+1}, \bar{s}_{n+2}, \ldots, \bar{s}_{\bar{n}}$ represent *spurious modes* due to parasitic energy-storage elements and have no counterpart in the correct solution.

The spurious modes are usually caused by high-frequency phase shift. In practice, the spurious-mode roots often have large negative real parts, so that the corresponding false-solution modes damp out very quickly and will not affect the solution appreciably. But in certain high-gain computing loops, or in "algebraic" feedback loops without integrators, phase shifts of cascaded amplifiers may add and cause spurious high-frequency oscillatory modes which render the computer setup useless. A suitable setup modification can often remedy such a situation by effectively damping the spurious high-frequency modes.

3-23. First-order Mode-error Calculations (see also Sec. 3-24). Although formulas permitting explicit computation of mode errors and spurious modes exist,[27–29] they are based on none-too-realistic standard computer setups and on greatly simplified formulations of computing-element frequency-response functions. The following simplified procedures often permit approximate mode-error calculations without restriction to a standard computer setup. In particular, the method of Sec. 3-23*b* permits comparisons of mode errors in different computer setups for the same problem.

(a) Low-frequency Errors.[27] In low-frequency computations, especially with inexpensive (low-gain) integrators, phase-shift errors are essentially due to imperfect integration at low frequencies (Secs. 1-10 and 3-20). Reference to Sec. 3-20 indicates that at low frequencies the transfer operator $-1/RCP$ of an ideal integrator is approximately replaced by $-1/RC(P + \lambda)$ with $\lambda = 1/r_{DC}C + 1/A_0RC$. The given differential equation (49) is then replaced by

$$F(P + \lambda)\bar{Y} \equiv [a_n(P + \lambda)^n + a_{n-1}(P + \lambda)^{n-1} + \cdots + a_0]\bar{Y} = 0 \qquad \left(P \equiv \frac{d}{d\tau}\right) \tag{3-53}$$

so that each real or complex root s_k of Eq. (49) is simply reduced by the same real quantity λ. The computer solution $\bar{Y}(\tau)$ is related to the desired correct solution $Y(\tau)$ by

$$\bar{Y}(\tau) = e^{-\lambda\tau}Y(\tau) \tag{3-54}$$

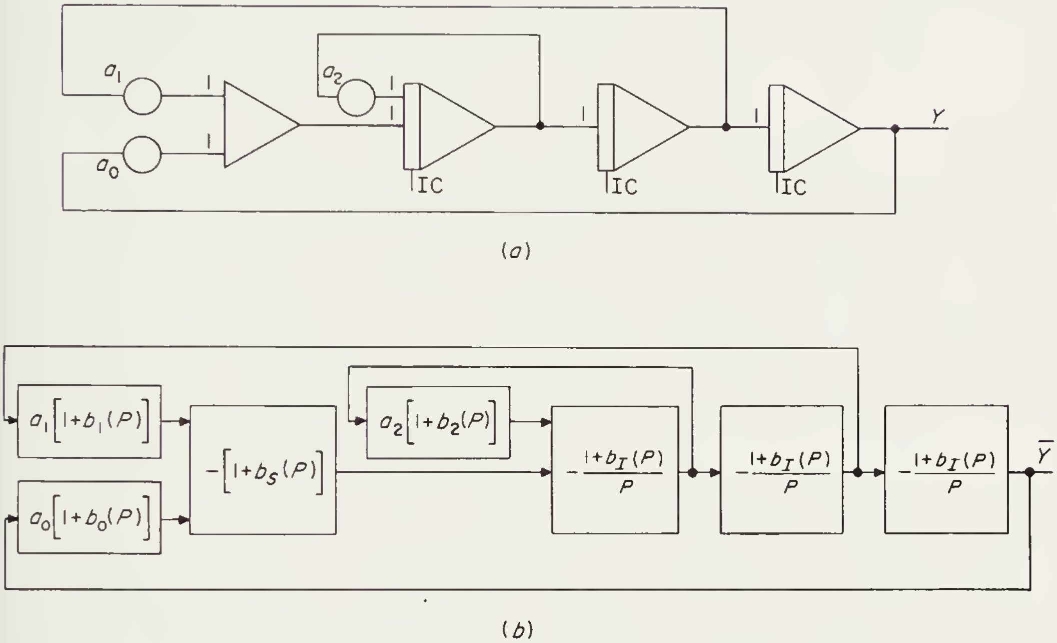
The resulting error could be corrected through multiplication by $e^{\lambda\tau}$ prior to recording of solutions.

(b) Assuming that the spurious modes can be neglected, the following procedure[26] permits approximate calculation of the mode changes. Let $\bar{F}(s) = F(s) + \Delta F(s)$ and $\bar{s}_k = s_k + \Delta s_k$, where $\Delta F(s)$ and the Δs_k are small. Then, since $F(s_k) = \bar{F}(\bar{s}_k) = 0$,

$$\bar{F}(s_k) = \Delta F(s_k) \approx -\bar{F}'(\bar{s}_k)\, \Delta s_k \tag{3-55}$$

so that the error Δs_k in the kth root s_k is approximately given by

$$\Delta s_k \approx -\frac{\Delta F(s_k)}{\bar{F}'(\bar{s}_k)} \approx -\frac{\Delta F(s_k)}{F'(s_k)} \tag{3-56}$$



(a)

(b)

Fig. 3-19. (*a*) Analog-computer setup for the differential equation

$$(P^3 + a_2P^2 + a_1P - a_0)Y = 0$$

and (*b*) a block diagram showing the effects of small linear errors.

Consider, for example, the computer setup of Fig. 3-19*a* for the differential equation

$$F(P)Y \equiv (P^3 + a_2P^2 + a_1P - a_o)Y = 0 \tag{3-57}$$

In Fig. 3-19*b*, the ideal integrator, summer, and potentiometer transfer operators $-1/P$, -1, and a_k have been replaced by the corresponding more realistic operators

$$-\frac{1}{P}[1 + b_I(P)] \qquad -[1 + b_S(P)] \qquad a_k[1 + b_k(P)]$$

where $b_I(P)$, $b_S(P)$, and $b_k(P)$ are small error terms respectively obtained from Secs. 3-20, 3-19, and 3-6. The actual differential equation implemented by the computer setup of Fig. 3-19 is

$$\bar{F}(P)\bar{Y} \equiv \left\{\frac{P^3}{(1+b_I)^3} + a_2(1+b_2)\frac{P^2}{(1+b_I)^2} + (1+b_S)\left[a_1(1+b_1)\frac{P}{1+b_I} - a_0(1+b_0)\right]\right\}\bar{Y} = 0 \tag{3-58}$$

If terms of the second and higher order in the various $b(P)$'s can be neglected, we have

$$\bar{F}(s) = F(s) + \Delta F(s) \approx F(s) - sb_I(s)F'(s) + b_S(s)(a_1 s - a_0) + a_2 b_2(s)s^2 + a_1 b_1(s)s - a_0 b_0(s)$$

and our first-order error estimate (56) for the root s_k is

$$\Delta s_k \approx s_k b_I(s_k) - \frac{b_S(s_k)(a_1 s_k - a_0) + a_2 b_2(s_k)s_k^2 + a_1 b_1(s_k)s_k - a_0 b_0(s_k)}{3s_k^2 + 2a_2 s_k + a_1} \tag{3-59}$$

An analogous procedure yields the root shifts Δs_k for every computer setup implementing a given differential equation (49) with integrators, summer/phase inverters, and coefficient potentiometers. *In each case, the first-order contribution of the integrator-frequency-response error to* Δs_k *will be* $+s_k b_I(s_k)$, *provided that the same error factor* $[1 + b_I(P)]$ *applies to every integration.*

3-24. Errors in Sinusoidal Modes. Circle Test and System Design. (a) The Circle Test. Mode-shift errors are most noticeable* when $\operatorname{Re} s_k = 0$ and Δs_k is real, i.e., when component errors cause a decay or increase in a supposedly sinusoidal (or constant) computer solution. The well-known computer setup for the harmonic-oscillator equation

$$F(P)Y \equiv (P^2 + \omega^2)Y = 0 \tag{3-60}$$

(Fig. 3-20*a*) demonstrates such a situation and serves as a sensitive test for phase shift in the computing elements used (*circle test*). Since $s_1 = j\omega$, $s_2 = -j\omega$, the correct solution for $Y(0) = B$, $\dot{Y}(0) = 0$ is

$$Y = B\cos\omega\tau \tag{3-61}$$

---

* This does *not* mean that similar component errors necessarily cause larger percentage errors in sinusoidal solutions. In the computer solution corresponding to Eq. (53), for instance, percentage errors are constant and the same for all modes; absolute errors increase or decrease with the solution. Errors in sinusoidal or constant solutions are simply easier to see, especially since such solutions can usually be observed for longer periods of time than increasing or decaying solutions.

Referring to Fig. 3-20*b*, the actual computer setup solves the machine equation

$$\bar{F}(P)\bar{Y} \equiv \left\{\frac{P^2}{[1+b_I(P)]^2} + \omega^2[1+b_S(P)][1+b_P(P)]\right\}\bar{Y} = 0 \tag{3-62}$$

where $b_I(P)$, $b_S(P)$, and $b_P(P)$ are small error terms due, respectively, to the integrator, phase inverter, and coefficient potentiometer, as in Sec. 3-23. Our first-order approximation leads to

$$\begin{aligned}\bar{F}(s) &= F(s) + \Delta F(s) \approx F(s) - 2s^2 b_I(s) + \omega^2[b_S(s) + b_P(s)]\\ F'(s) &= 2s\end{aligned}$$

and, using Eq. (56),

$$\Delta s_k \approx s_k b_I(s_k) - \frac{\omega_2}{2s_k}[b_S(s_k) + b_P(s_k)] \tag{3-63}$$

where $s_k = \pm j\omega$. Since $b_I(P)$, $b_S(P)$, and $b_P(P)$ are real transfer operators, we must have $b(-j\omega) = -b(j\omega)$ for each b in Eq. (63), so that

$$\Delta s_k = j\omega\left[b_I(j\omega) + \frac{b_S(j\omega) + b_P(j\omega)}{2}\right] \tag{3-64}$$

for both $s_1 = j\omega$ and $s_2 = -j\omega$.

Expressions for $b_I(j\omega)$, $b_S(j\omega)$, and $b_P(j\omega)$ in terms of circuit parameters are readily obtained from Secs. 3-20, 3-19, and 3-6; these quantities can also be obtained directly from frequency-response data as complex (phasor) fractional component errors for sinusoidal excitation.

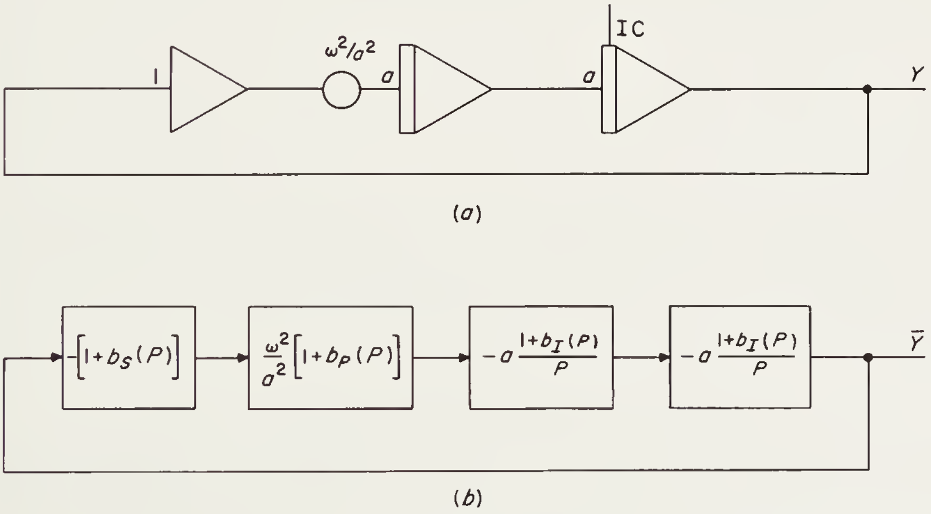


Fig. 3-20. (*a*) A circle-test setup implementing $(P^2 + \omega^2)Y = 0$, and (*b*) a block diagram showing the effects of small linear errors.

Assuming accurate gain calibration, component errors will be practically entirely due to phase shift at all frequencies where our first-order analysis applies, so that

$$b_I(j\omega) = j\delta_I(\omega) \qquad b_S(j\omega) = j\delta_S(\omega) \qquad b_P(j\omega) = j\delta_P(\omega) \quad (3\text{-}65)$$

where $\delta_I(\omega)$, $\delta_S(\omega)$, and $\delta_P(\omega)$ are simply the component phase errors (obtained from Secs. 3-20, 3-19, and 3-6, or measured; see also Sec. 3-24*b*). Hence the root shift

$$\Delta s_k = -\omega\left[\delta_I(\omega) + \frac{\delta_S(\omega) + \delta_P(\omega)}{2}\right] = -\omega\delta(\omega) \qquad (3\text{-}66)$$

is real and equals $-\omega$ times one-half the total phase shift $2\delta(\omega)$ around our computer loop. The resulting shifted roots $\bar{s}_k$ obtained from our first-order theory are

$$\bar{s}_{1,2} = \pm j\omega - \omega\left[\delta_I(\omega) + \frac{\delta_S(\omega) + \delta_P(\omega)}{2}\right] = \pm j\omega - \omega\delta(\omega) \quad (3\text{-}67)$$

The correct solution $Y(\tau) = B \cos \omega\tau$ *of Eq.* (60) *is, then, replaced by the actual computer solution*

$$\bar{Y}(\tau) = Be^{-\omega\delta(\omega)\tau} \cos \omega\tau \qquad (3\text{-}68)$$

The solution amplitude decays exponentially with time for a positive net phase error $2\delta(\omega) = 2\delta_I(\omega) + \delta_S(\omega) + \delta_P(\omega)$ *(lead), while a negative net phase error (lag) will cause the amplitude to increase.* The fractional error in the *n*th maximum ($\tau = 2\pi n/\omega$, $n = 1, 2, \ldots$) of $\bar{Y}(\tau)$, i.e.,

$$\frac{1}{B}\left[B - \bar{Y}\left(\frac{2\pi n}{\omega}\right)\right] = 1 - e^{-2\pi n\delta(\omega)} \approx 2\pi n\delta(\omega) \qquad (n = 1,2, \ldots) \quad (3\text{-}69)$$

is easy to measure and permits convenient study of the various sources of phase shift.

(b) Examples and System-design Considerations. Reference to Secs. 3-6, 3-19, and 3-20 shows that for suitably small errors (component phase errors below 10 deg < 0.2 radian are reasonable)

$$\delta(\omega) = \delta_I(\omega) + \tfrac{1}{2}\delta_S(\omega) + \tfrac{1}{2}\delta_P(\omega) \approx \frac{1}{\omega A_0 RC_{\text{EFF}}} + D(\omega)$$
$$+ \omega RC_R - \frac{\delta_A(\omega)}{A_0}\left(1 + \frac{C_G}{C_{\text{EFF}}}\right) + \tfrac{1}{2}\delta_S(\omega) + \tfrac{1}{2}\delta_P(\omega) \quad (3\text{-}70)$$

where A_0 and $\delta_A(\omega)$ are assumed to be the same for all three amplifiers.

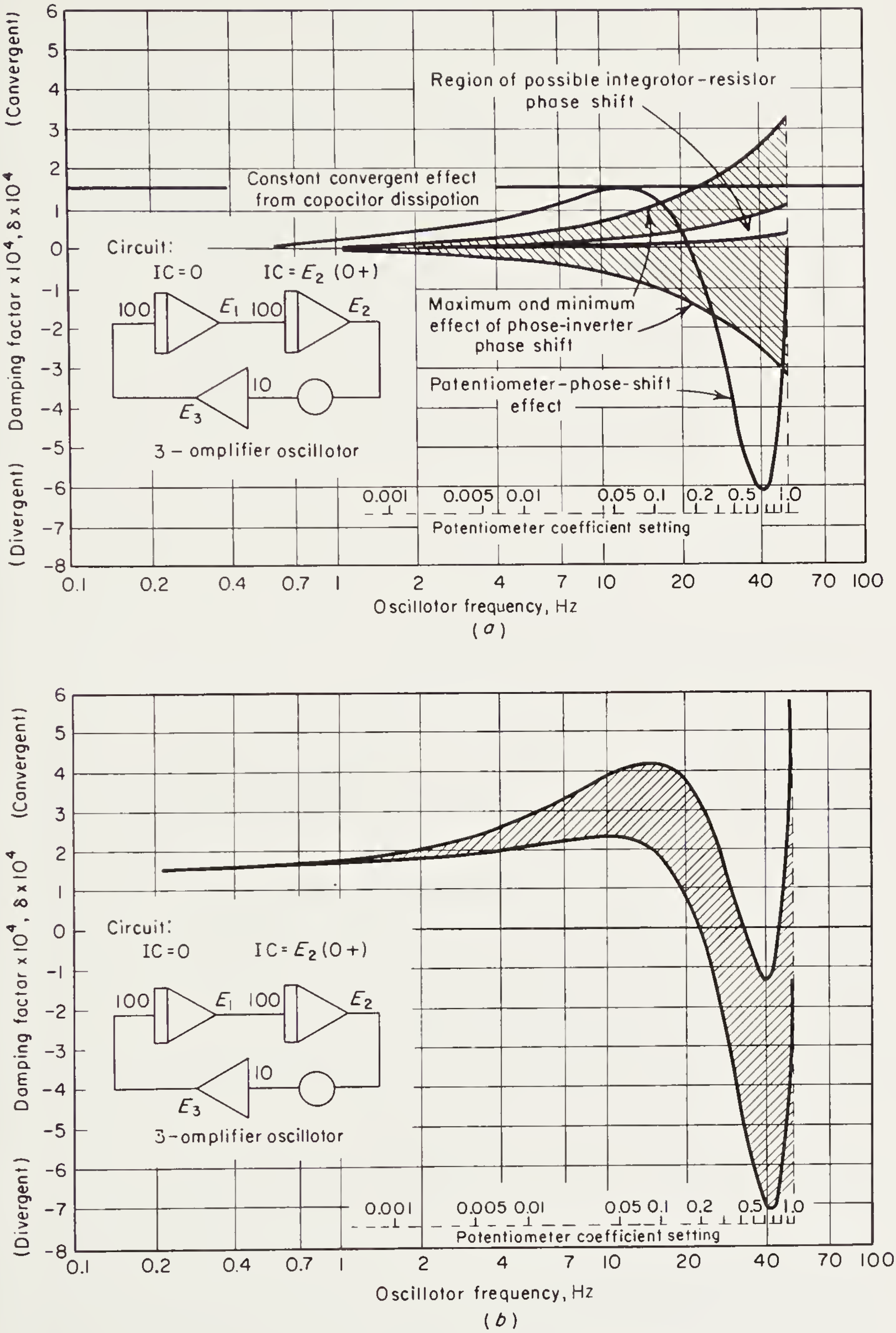


Fig. 3-21 *a* and *b*. (*a*) Individual effects and (*b* to *e*) combined effect of various sources of phase shift on circle-test damping. (*Based on Refs.* 13 *and* 14; *a fairly slow* ±100-*volt machine was used.*) Note that $\delta = 8 \times 10^{-4}$ corresponds to approximately 0.5 percent amplitude decay per oscillation cycle.

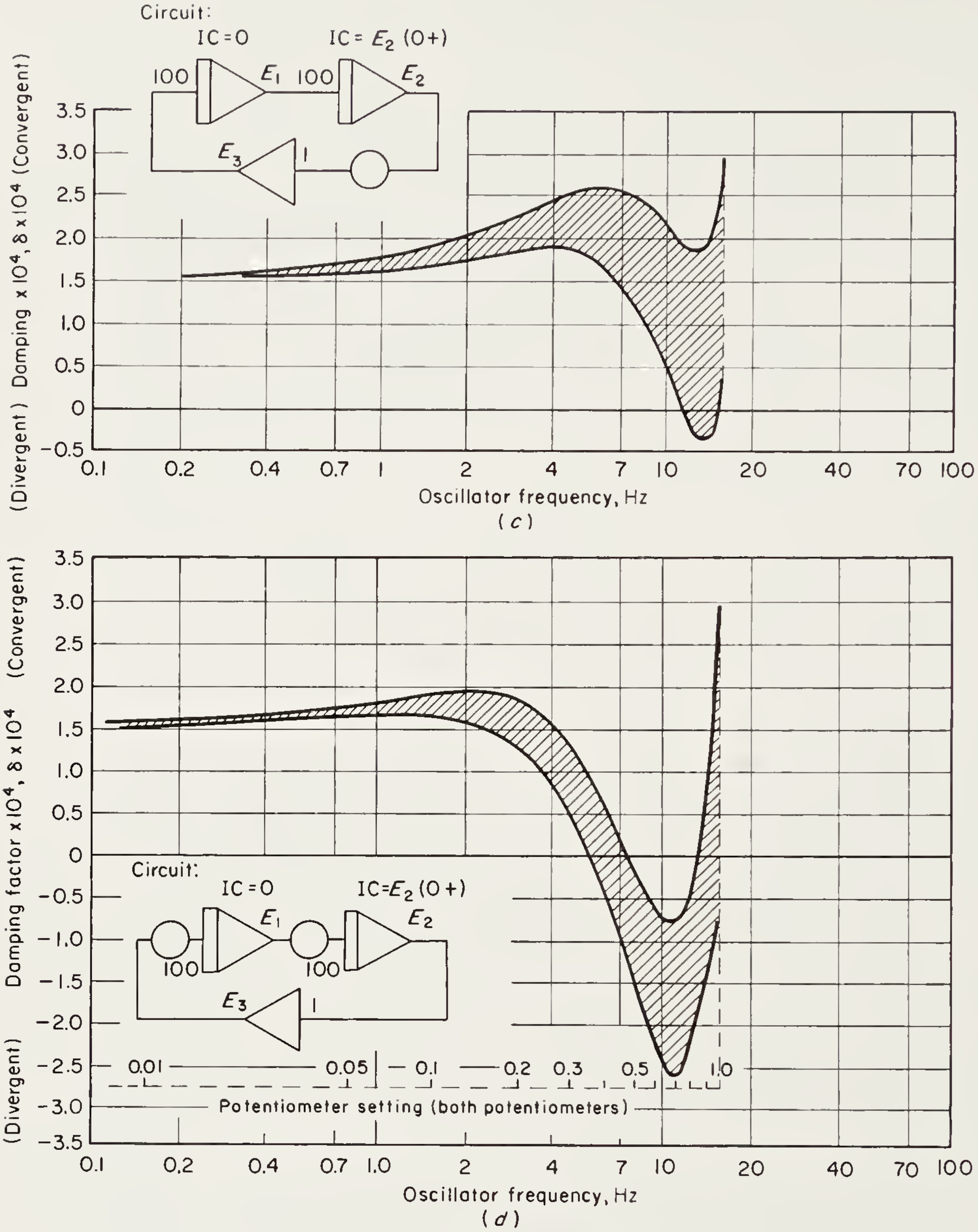
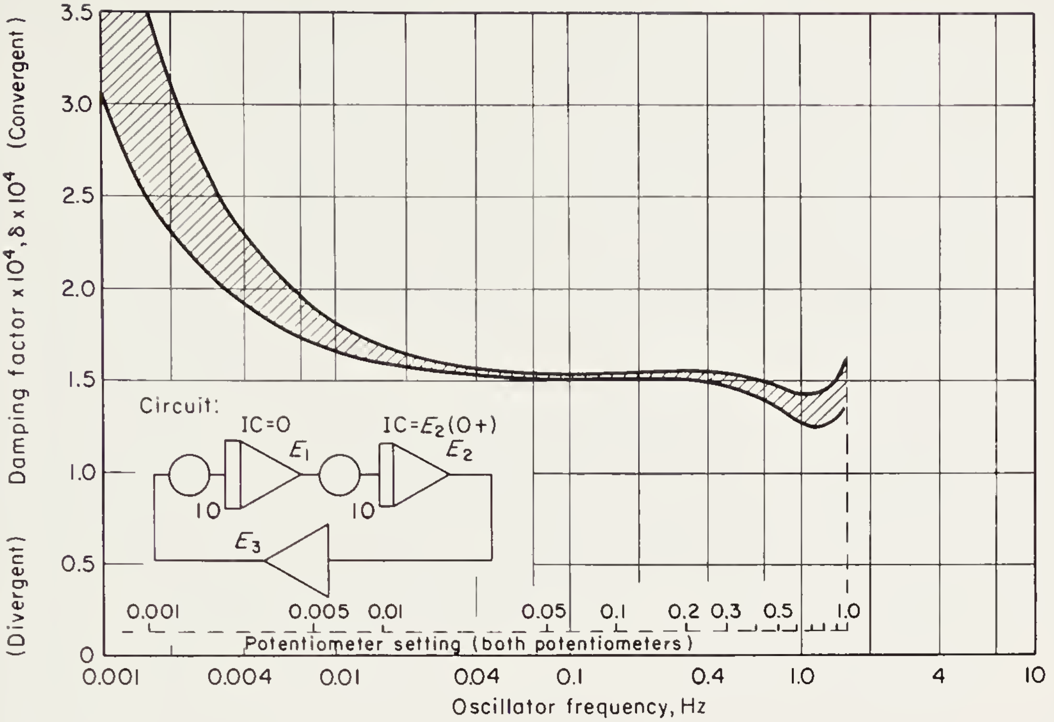


Fig. 3-21*c and d*

The potentiometer phase shift $\delta_P(\omega)$ can be found from Fig. 3-9 for the correct potentiometer setting α.

Note that the sum (70) for the damping ratio $\delta(\omega)$ can contain negative as well as positive terms. Figure 3-21*a* illustrates the individual contributions of various error sources for a typical circle-test setup as the oscillation frequency $\omega/2\pi$ is changed by the potentiometer setting, and



(*e*)

Fig. 3-21*e*

Fig. 3-21*b* indicates their combined effect. Figure 3-22 shows similar plots of $\delta(\omega)$ for different combinations of integrator input gain and phase-inverter gain. Finally, Table 3-2 summarizes the effects of various error sources in "slow" and wideband electronic integrators.

Circle-test data taken over a suitable frequency range appear to give a fairly conservative indication of combined error effects on computer solutions of linear if not nonlinear differential equations. The data of Fig. 3-21 were taken with a relatively slow electronic differential analyzer designed for moderate-speed real-time flight simulation. Figure 3-22 applies to a very fast analog/hybrid computer suitable for iterative computation at up to 4,000 solutions per second. Below 0.1 Hz, $\delta(\omega) = 1/r_{DC}C + 1/A_0RC$.

With high-quality equipment ($r_{DC}C > 5 \times 10^5$ sec, $A_0 \geq 10^7$), the resulting one-cycle decay is between 0.05 and 0.1 percent. At higher frequencies (100 Hz to 1 MHz, repetitive/iterative and hybrid computation), errors begin to be affected by d-c amplifier frequency response (δ_A/A_0 terms), but this effect is usually masked or compensated by phase shift determined by C_R, C_0, and C_1, and especially by potentiometer phase shift (Table 3-2). At these higher frequencies, phase inverters, summers, and potentiometers must employ lower resistance values and be carefully phase-compensated, and *distributed capacitances to ground*

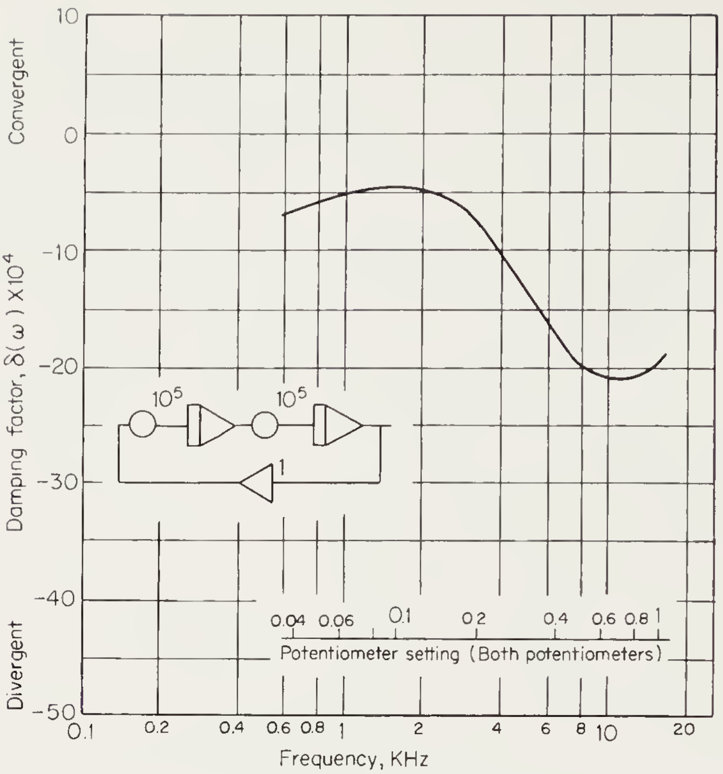


Fig. 3-22. Circle-test data for a fast ±10-volt iterative differential analyzer with low-impedance computing networks ($R = 1\text{K}$, $C = 0.01$ μF, potentiometer resistance = 2K). Amplifiers have unity-gain bandwidth of 30 MHz. (*University of Arizona LOCUST computer, Ref.* 65.)

can become critical. These problems suggest the use of low-impedance, high-current transistor amplifiers and miniaturized construction with short wiring runs (Sec. 7-2*b*).

DIFFERENTIAL AND NONINVERTING OPERATIONAL AMPLIFIERS

3-25. Introduction. Many d-c amplifiers are actually *differential amplifiers* having a *noninverting input* as well as the *inverting input* used in "classical" operational amplifiers (Sec. 3-10). With accurately matched inverting and noninverting gains (high common-mode rejection, Sec. 3-26), *both* amplifier inputs can be used for computation, so that a much wider variety of operational amplifiers can be designed (Sec. 3-27). The application of differential amplifiers to *noise cancellation* in instrumentation and data-transmission systems is even more important (see also Sec. 7-14).

3-26. Differential and Common-mode Amplification. The input voltages E_1, E_2 of the differential amplifier shown in Fig. 3-23*a* can be expressed in the form

$$E_1 = \frac{E_1 - E_2}{2} + \frac{E_1 + E_2}{2} = \frac{E_1 - E_2}{2} + E_{CM}$$

$$E_2 = -\frac{E_1 - E_2}{2} + \frac{E_1 + E_2}{2} = -\frac{E_1 - E_2}{2} + E_{CM} \qquad (3\text{-}71)$$

where the *difference* $E_1 - E_2$ constitutes a *useful* signal to be amplified, while the *common-mode input* $E_{CM} = (E_1 + E_2)/2$ is an *undesirable* noise, hum, or d-c drift voltage common to both inputs. The amplifier output will be

$$X_0 = A_1E_1 - A_2E_2 \qquad (3\text{-}72)$$

where the voltage gains A_1, A_2 may be regarded as frequency-dependent phasor operators for single-frequency sinusoidal input components. In general, A_1 and A_2 are real and *negative* at d-c. In an *ideal* differential amplifier, $A_1 = A_2$, so that X_0 is truly proportional to the input difference $E_1 - E_2$; the common-mode input E_{CM} is completely rejected. In general, this ideal situation is only approximated more or less closely, and we have

$$X_0 = A_1E_1 - A_2E_2 = G_D(E_1 - E_2) + G_{CM}E_{CM}$$

$$= G_D\left(E_1 - E_2 + \frac{1}{CMRR}\frac{E_1 + E_2}{2}\right) \qquad (3\text{-}73)$$

with

$$G_D = \frac{A_1 + A_2}{2} \qquad \text{(DIFFERENTIAL GAIN)} \qquad (3\text{-}74)$$

$$G_{CM} = \frac{G_D}{CMRR} = A_1 - A_2 \qquad \text{(COMMON-MODE GAIN)} \qquad (3\text{-}75)^*$$

$$CMRR = \frac{G_D}{G_{CM}} = \frac{1}{2}\frac{A_1 + A_2}{A_1 - A_2} \qquad \text{(COMMON-MODE REJECTION RATIO)} \qquad (3\text{-}76)$$

The *equivalent common-mode input* $E_{CM}/CMRR$ in Eq. (73) equals the equivalent differential error voltage which would cause the undesirable common-mode output $G_D(E_1 + E_2)/2CMRR$ in an ideal differential amplifier with differential gain G_D (Fig. 3-23*b*). The *common-mode rejection ratio CMRR is, therefore, an important measure of differential-amplifier quality.*

High $|CMRR|$ values (10^3 to 10^6 at d-c) are readily obtained through combinations of transistor matching, monolithic amplifier construction, cascaded differential-amplifier stages, and *common-mode feedback* around the amplifier input stage or around several stages.[74–80] *Note that |CMRR| is a function of frequency* and will, in general, decrease with frequency; unfortunately, many manufacturers specify |CMRR| only at d-c or at 100 Hz.

---

* Equations (73) and (75) define *CMRR* as a signed quantity. Amplifier specifications usually quote $|CMRR|$ (or 20 $\log_{10} |CMRR|$ in decibels).

With finite-impedance sources, undesirable common-mode output can occur even with infinite |CMRR| because of differences in the source impedances and amplifier input impedances to ground. These effects are minimized by amplifiers having very high input impedances (FET input), but even then stray shunt (leakage) resistances and capacitances to grounds, shields, or power supplies can cause unbalance effects (see the bibliography for Chap. 7).

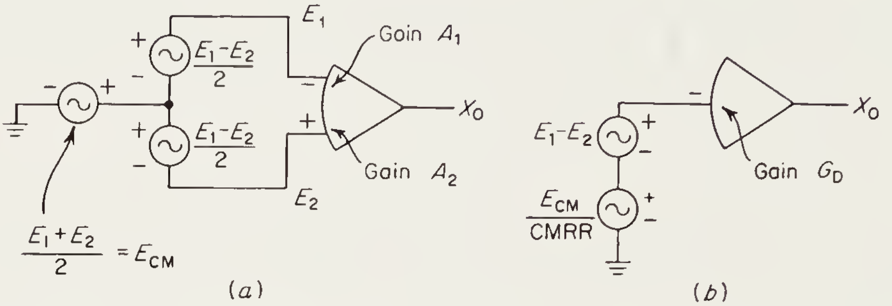


Fig. 3-23. Differential amplifier with equivalent sources (*a*) and equivalent circuit (*b*).

3-27. Differential and Noninverting Operational Amplifiers. Figure 3-24 shows a general differential-input operational-amplifier circuit. We first omit the amplifier-offset terms E_{d1}, E_{d2}, i_{d1}, i_{d2}. The circuit is described by three simple node equations

$$E_1 = \alpha_1 X_1 + \gamma_1 X_2 + \beta_1 X_0 \tag{3-77a}$$

$$E_2 = \alpha_2 X_2 + \gamma_2 X_1 + \beta_2 X_0 \tag{3-77b}$$

$$X_0 = A_1 E_1 - A_2 E_2 \tag{3-77c}$$

$\alpha_1, \alpha_2, \gamma_1, \gamma_2, \beta_1, \beta_2, A_1$, and A_2 are frequency-dependent voltage transfer functions determined by the details of the input and feedback networks shown; the derivation of these transfer functions in specific cases will be discussed in Sec. 3-28. Straightforward solution of Eq. (77) yields the *exact performance equation*

$$X_0 = \frac{(A_1\alpha_1 - A_2\gamma_2)X_1 - (A_2\alpha_2 - A_1\gamma_1)X_2}{1 - (A_1\beta_1 - A_2\beta_2)} \tag{3-78}$$

The net loop gain $A_1\beta_1 - A_2\beta_2$ must satisfy the usual stability conditions (see also Sec. 3-34) and, for proper operational-amplifier operation (performance insensitive to amplifier gain), *the net degenerative loop gain must be large;* i.e.,

$$|A_1\beta_1 - A_2\beta_2| \gg 1 \tag{3-79a}$$

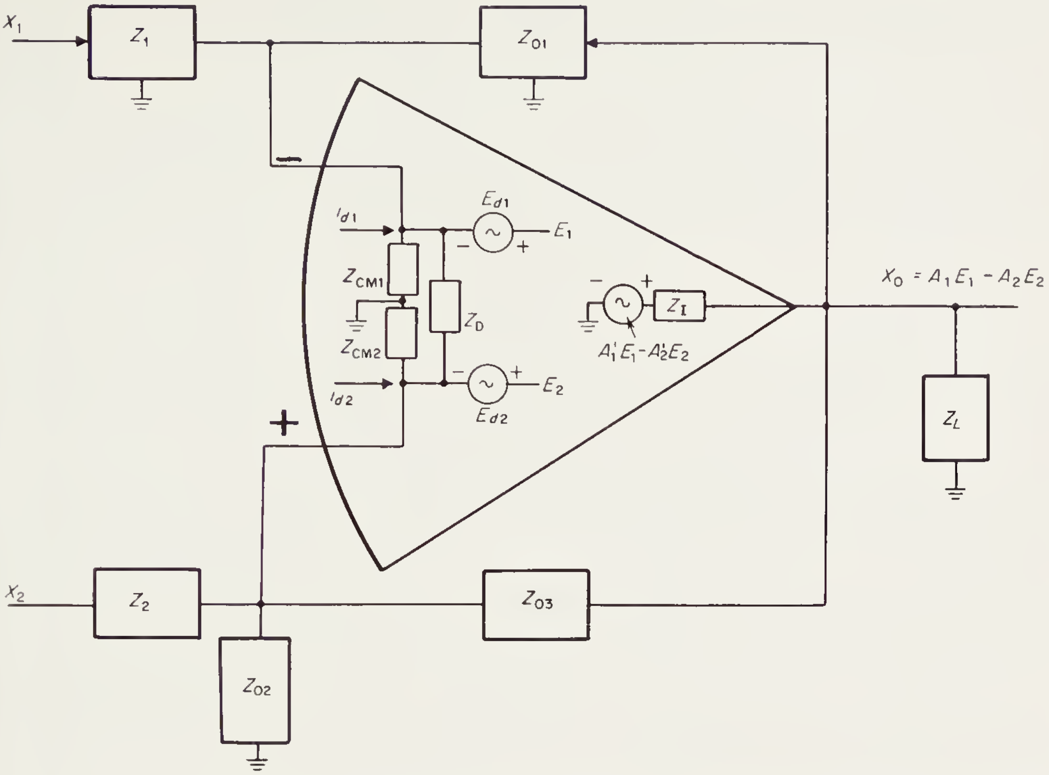


Fig. 3-24. An operational amplifier with inverting and noninverting inputs. Inverting and noninverting operational amplifiers are special cases of this circuit, which can be further generalized through addition of several input currents at the inverting and noninverting summing points, as in Fig. 3-15. Note that Z_1 and Z_{01} can be four-terminal transfer impedances, while Z_2, Z_{02}, Z_{03}, Z_{CM1}, Z_{CM2}, Z_D, Z_I, Z_L are two-terminal impedances.

In practical circuits with high common-mode rejection,

$$A_2 = A_1 \frac{1 - \dfrac{1}{2CMRR}}{1 + \dfrac{1}{2CMRR}} \approx A_1\left(1 - \frac{1}{CMRR}\right) \approx A_1 \qquad \text{(3-79}b\text{)}$$

Again, the parasitic *differential input impedance* Z_D of practical amplifiers will be large compared to other circuit impedances at working frequencies, so that

$$|\gamma_1| \ll |\alpha_1| \qquad |\gamma_2| \ll |\alpha_2| \qquad \text{(3-79}c\text{)}$$

If the relations (79) hold so that we can neglect all but first-order terms in $1/(A_1\beta_1 - A_2\beta_2)$, $1/CMRR$, γ_1, and γ_2, Eq. (78) leads to

$$X_0 \approx -\frac{\alpha_1X_1 - \alpha_2X_2}{\beta_1 - \beta_2}\left[1 + \frac{1}{A_1(\beta_1 - \beta_2)} - \frac{\beta_2}{(\beta_1 - \beta_2)CMRR}\right] + \frac{\gamma_2X_1 - \gamma_1X_2}{\beta_1 - \beta_2} - \frac{\alpha_2}{(\beta_1 - \beta_2)CMRR}X_2 \qquad \text{(3-80)}$$

which approximates the *idealized performance equation*

$$X_0 \approx -\frac{\alpha_1 X_1 - \alpha_2 X_2}{\beta_1 - \beta_2} \tag{3-81}$$

The amplifier *closed-loop output impedance* Z_S is derived from Eq. (78) just as in Sec. 3-11, with the approximate result

$$Z_S \approx -\frac{Z_I}{A'_1\beta_1 - A'_2\beta_2} \tag{3-82}$$

based on the approximations (79), where the *open-loop source impedance* Z_I and the *no-load open-loop gains* A'_1, A'_2 are defined in Fig. 3-24. *Errors due to gain changes or distortion* are treated exactly as in Sec. 3-16.

3-28. Important Special Cases. **(a)** Figure 3-25 shows several examples of *single-input operational amplifiers employing both the inverting and noninverting connections.* Other examples are the *negative-immittance converters* used in some active-filter designs.[93]

(b) More frequently, only *degenerative* feedback is present, i.e., $\beta_2 = 0$. For $X_2 = 0$, the circuit must, then, act as an ordinary inverting operational amplifier. Therefore, α_1/β_1 is, as in Sec. 3-11, the ratio of the short-circuit transfer impedances Z_{01} and Z_1:

$$\frac{\alpha_1}{\beta_1} = \frac{Z_{01}}{Z_1} \tag{3-83a}$$

Assuming that the parasitic input impedances Z_D, Z_{CM1}, Z_{CM2} are so large that they do not noticeably affect α_2 and β_1 at working frequencies,

$$\alpha_2 \approx \frac{Z_{02}}{Z_2 + Z_{02}} \tag{3-83b}$$

Figure 3-26 shows examples.

(c) In a *differential operational amplifier* (Fig. 3-26), we will assume that α_2 is given by Eq. (3-83b), and that Z_1 and Z_{02} are also two-terminal impedances, with

$$\beta_1 \approx \frac{Z_1}{Z_1 + Z_{01}} \tag{3-84}$$

We now choose $Z_{02}/Z_2 = Z_{01}/Z_1$, so that Eqs. (80) and (83) yield

$$\left.\begin{aligned} X_0 &\approx -\frac{Z_{01}}{Z_1}(X_1 - X_2)\left(1 + \frac{1}{A_1\beta_1}\right) + \frac{\gamma_2 X_1 - \gamma_1 X_2}{\beta_1} \\ &\qquad - \frac{Z_{01}}{Z_1}\frac{1}{CMRR}X_2 \\ &\approx -\frac{Z_{01}}{Z_1}(X_1 - X_2) \end{aligned}\right\} \tag{3-85}$$

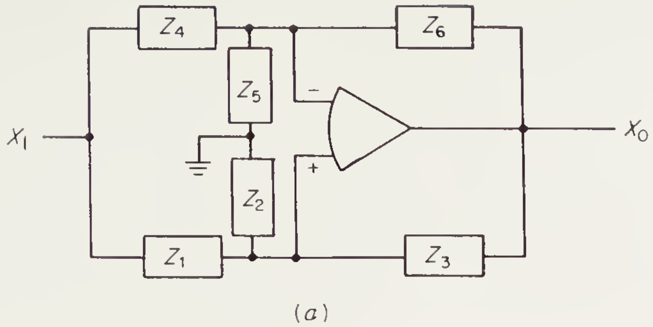


Fig. 3-25*a*. Single-input operational amplifier employing both the inverting and non-inverting connections. In terms of the two-terminal admittances $Y_K = 1/Z_K$,

$$X_0 = \frac{Y_1(Y_4 + Y_5 + Y_6) - Y_4(Y_1 + Y_2 + Y_3)}{Y_6(Y_1 + Y_2 + Y_3) - Y_3(Y_4 + Y_5 + Y_6)} X_1$$

Unfortunately, the subtractions usually necessitate very close component tolerances. One often chooses $Y_4 + Y_5 + Y_6 = Y_1 + Y_2 + Y_3$, so that

$$X_0 = \frac{Y_1 - Y_4}{Y_6 - Y_3} X_1$$

(*Based on Ref.* 93.)

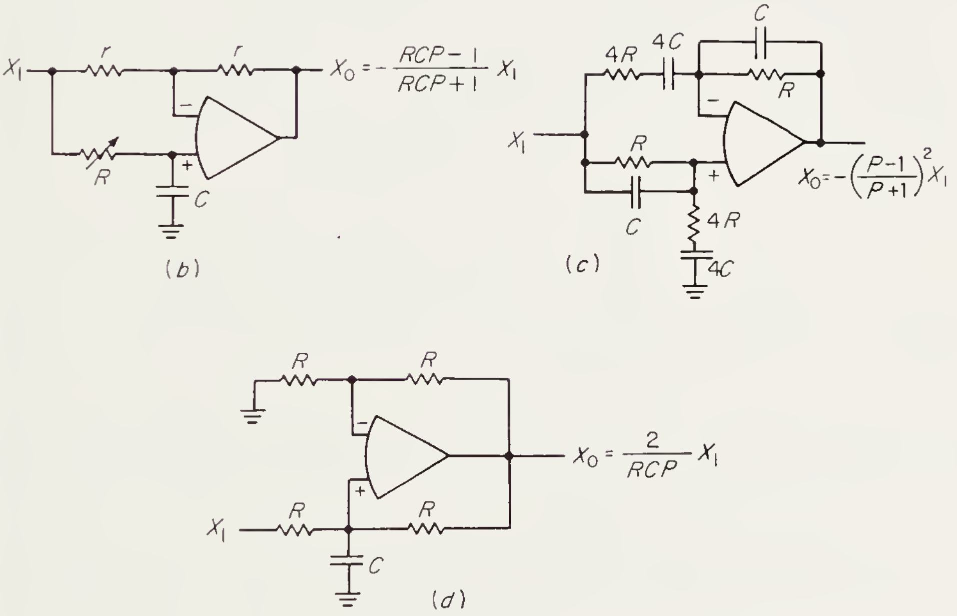

X_1
r
r
$X_0 = -\frac{RCP-1}{RCP+1} X_1$
−
+
R
C
(b)
C
4R
4C
R
−
X_1
R
+
$X_0 = -\left(\frac{P-1}{P+1}\right)^2 X_1$
C
4R
4C
(c)
R
R
−
+
$X_0 = \frac{2}{RCP} X_1$
X_1
R
R
C
(d)

Fig. 3-25*b* to *d*. Two all-pass phase shifters, (*b*) and (*c*), and a noninverting integrator (*d*). Accurate component matching is required.

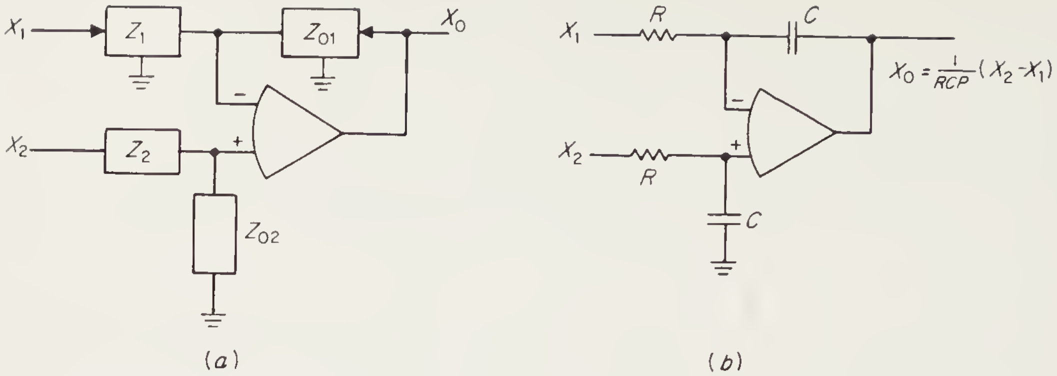


Fig. 3-26. Differential operational amplifier (*a*) and differential integrator (*b*).

Figure 3-26*b* shows an example. Equation (85) permits estimation of errors due to finite gain, *CMRR*, and Z_D, provided that Z_D, Z_{CM1}, and Z_{CM2} are sufficiently large to ensure (83*b*) and (84).

(d) In a *noninverting operational amplifier*, $X_1 = 0$. For high-impedance input (see also Sec. 3-31), one usually makes $Z_{02} = \infty$. Figure 3-27*a* shows a typical example. To account for the parasitic input impedances in Fig. 3-27*a*, assume that Z_D and Z_{CM1} are so large that the feedback ratio β_1 is given by Eq. (84) and $\alpha_2 \approx 1$ but that we cannot neglect

$$\gamma_1 \approx \frac{Z_0 Z_1}{Z_0 Z_1 + (Z_0 + Z_1) Z_D} \approx \frac{Z_0 Z_1}{(Z_0 + Z_1) Z_D} \tag{3-86}$$

Equation (80) with $X_1 = 0$, $\beta_2 = 0$ leads us to

$$X_0 \approx \frac{1}{\beta_1}\left(1 + \frac{1}{A_1 \beta_1} - \frac{1}{CMRR} - \gamma_1\right) X_2$$

with

$$\beta_1 \approx \frac{Z_1}{Z_0 + Z_1} \tag{3-87}$$

which should be compared to the ideal result $X_0 = X_2/\beta_1$. In particular, Eq. (87) yields the approximate output, including first-order errors,

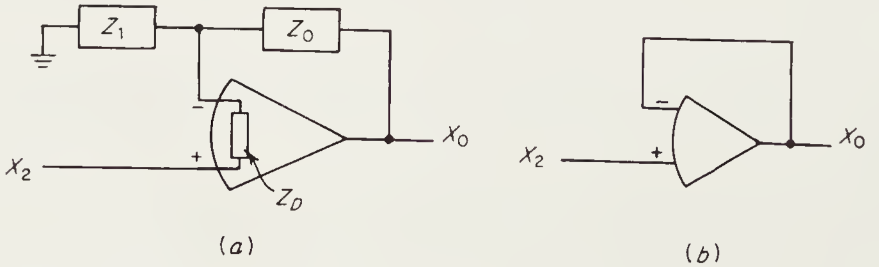


Fig. 3-27. Noninverting operational amplifier (*a*) and unity-gain follower (*b*).

of the important *unity-gain-follower circuit* shown in Fig. 3-27*b* ($\beta_1 = 1$):

$$X_0 \approx \left(1 + \frac{1}{A_1} - \frac{1}{CMRR}\right) X_2 \approx X_2 \tag{3-88}$$

3-29. Effects of D-c Offset and Noise. It remains to estimate the effects of d-c offset (or noise) voltages E_{d1}, E_{d2}, i_{d1}, i_{d2} linearly superimposed on our results for Fig. 3-24 (see also Sec. 3-15). For sufficiently large loop gain, $|CMRR|$, and parasitic input impedances, the resulting error voltage is given by

$$e_d \approx \frac{-1}{\beta_1 - \beta_2} [(E_{d1} - E_{d2}) + (Z_{s1} i_{d1} - Z_{s2} i_{d2})] \tag{3-89}$$

where Z_{s1} and Z_{s2} are the respective combined shunt impedances from the two amplifier inputs to ground, source, and/or output. If Z_{01} is a two-terminal impedance, and $\beta_2 = 0$, note that

$$\frac{Z_{s1}}{\beta_1} = Z_{01} \tag{3-90}$$

For the unity-gain follower of Fig. 3-27*b*, Eq. (89) yields

$$e_d \approx -(E_{d1} - E_{d2}) + Z_{s2} i_{d2} \tag{3-91}$$

Note carefully that the d-c offset specifications crucial for Eqs. (89) and (91) are the *differential* voltage offset $E_{d1} - E_{d2}$ and the *single-ended* (bias-) current offsets i_{d1}, i_{d2}. Only for symmetrical differential circuits like that in Fig. 3-25*b* will Eq. (89) reduce to the more favorable relation

$$e_d \approx -\frac{1}{\beta_1}(E_{d1} - E_{d2}) - Z_{01}(i_{d1} - i_{d2}) \tag{3-92}$$

3-30. Input Impedances (see also Sec. 3-14*b*). **(a)** In Fig. 3-28, $\beta_1 = \beta$, $\beta_2 = 0$, and Z_1 is a two-terminal impedance. The source of the

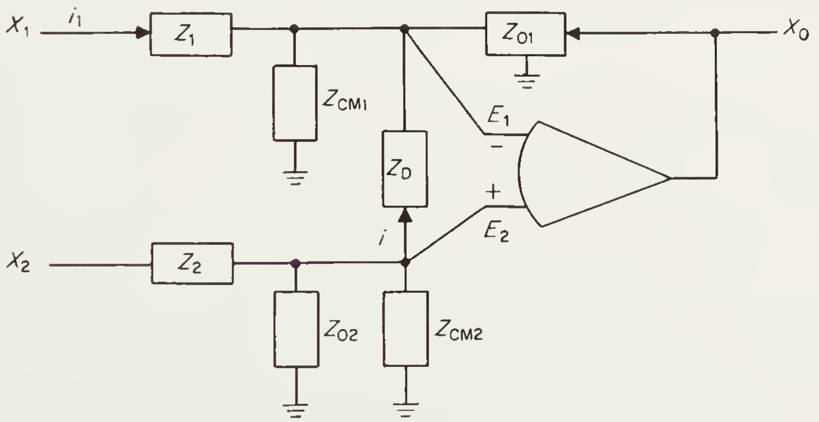


Fig. 3-28. Input-impedance calculation.

input voltage X_1 sees the impedance Z_1 grounded in series with the voltage E_1. For $X_2 = 0$, $X_0 \approx -Z_{01}X_1/Z_1$, and Eqs. (77) and (79*b*) lead to

$$E_1 = \frac{X_0}{A_1} + \frac{A_2}{A_1}E_2 \approx -X_1\left[\frac{Z_{01}}{A_1Z_1} - \gamma_2\left(1 - \frac{1}{CMRR}\right)\right]$$

Now $i_1 = (X_1 - E_1)/Z_1$, and the X_1 source sees the impedance

$$\begin{aligned} Z_1' = \frac{X_1}{i_1} &\approx \frac{Z_1}{1 + Z_{01}/A_1Z_1 - \gamma_2(1 - 1/CMRR)} \\ &\approx Z_1\left(1 - \frac{Z_{01}}{A_1Z_1} + \gamma_2\right) \end{aligned} \tag{3-93}$$

which is usually well approximated by Z_1, as in a simple noninverting operational amplifier.

(b) Refer again to Fig. 3-28, and let $X_1 = 0$. *The input impedance seen at the noninverting amplifier terminal* will be Z_{02} and Z_{CM2} in parallel with

$$Z = \frac{E_2}{i} \qquad \text{with} \qquad i = \frac{E_2 - E_1}{Z_D}$$

Now Eq. (77) yields

$$\begin{aligned} E_1 &= \frac{E_2}{1 - A_1\beta}\left[\frac{\gamma_1}{\alpha_2} - A_1\beta\left(1 - \frac{1}{CMRR}\right)\right] \\ &\approx E_2\left(1 - \frac{1 - \gamma_1/\alpha_2}{A_1\beta} - \frac{1}{CMRR}\right) \end{aligned} \tag{3-94}$$

so that

$$Z \approx \frac{Z_D}{\dfrac{1 - \gamma_1/\alpha_2}{A_1\beta} + \dfrac{1}{CMRR}}$$

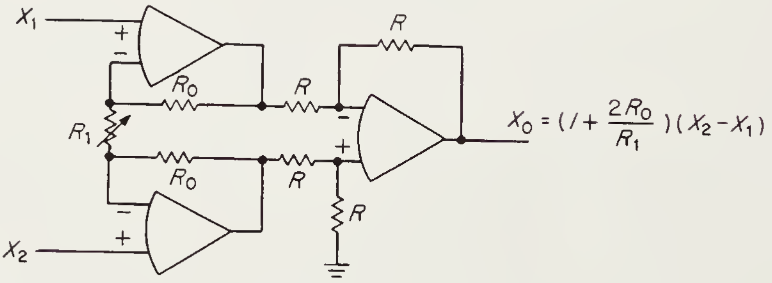


Fig. 3-29. An instrumentation amplifier with high-impedance differential inputs.

which can be very large for large $|A_1\beta|$ and $|CMRR|$. In particular, *for the unity-gain follower of Fig.* 3-27*b*, $\beta = \alpha_2 = 1$, and $\gamma_1 = 0$, so that

$$Z \approx \frac{Z_D}{1/A_1 + 1/CMRR} \tag{3-95}$$

(*high-impedance-follower circuit*).

Figure 3-29 shows the design of a high-impedance-input difference amplifier for instrumentation.

D-C AMPLIFIER SPECIFICATIONS, DESIGN, AND TESTS

3-31. Survey. Tables 3-3 and 3-4 illustrate important specifications for computer d-c amplifiers (see also Sec. 3-37). Integrated-circuit amplifiers will, in general, only supply 2 to 10 mA of output current at ±10 volts, but many discrete-component amplifiers combine such integrated circuits with powerful output stages (Fig. 3-35). Figure 3-30 shows some typical amplifiers.

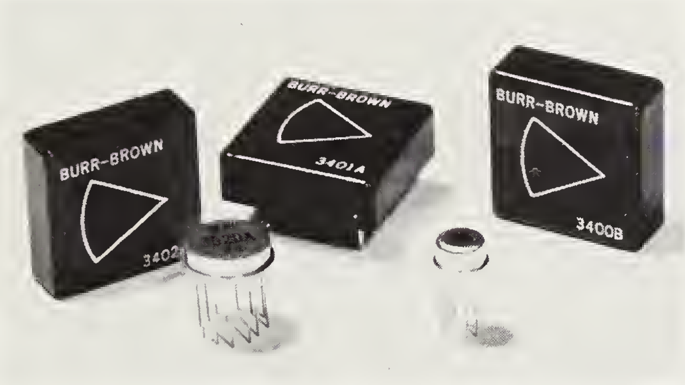


Fig. 3-30. Computer d-c amplifiers: integrated circuits and epoxy-cast modules. (*Burr-Brown Research Corporation.*)

3-32. Drift Specifications. D-c amplifier drift is specified in terms of the *equivalent voltage and current offset* after a given time interval, power-supply-voltage change, or temperature change. The equivalent-input-offset voltage E_d and current i_d are respectively defined as minus the *input* voltage and current needed to return the open-loop output to zero (balance condition) with a grounded voltage source and a current source at the summing point (see also Secs. 3-15, 3-30, 3-33, and Fig. 3-31). *Unless the source resistances at the inverting and non-inverting inputs of a differential amplifier are matched* (*and this is often not possible*), *current drift is determined by the total current drift at a single input* ("*bias-current*" *drift*), *rather than by the smaller differential current* *drift often quoted by manufacturers.* Table 3-5 illustrates drift-error effects for different types of computer amplifiers operated as summers and integrators (or sample-hold circuits, Sec. 5-18) over normal laboratory temperature ranges. Errors over wider temperature ranges can be calculated with the aid of Eqs. (33), (34), and (48).

3-33. Low-drift Amplifier Design. A *differential input stage* with a current-source transistor in the emitter return (Fig. 3-32*a*) ties the temperature dependence of the first-stage output to the *difference* of two transistor V_{BE}'s; transistor matching for V_{BE} and β can reduce voltage drift from 2.5 mV/deg C to below 5 μV/deg C. Matching is especially favorable in monolithic integrated circuits; differential heating of the two input transistors must be minimized by low values of the collector current ($<$200 μA). The first-stage gain should be sufficiently high (at least 10) to make drift effects in later stages negligible (Fig. 3-31). A roughly matched second differential-amplifier stage may follow the first for more perfect common-mode rejection, or common-mode feedback from a succeeding stage may be used.

We note in passing that differential-amplifier voltage drift can be reduced below 0.5 μV/deg C if the V_{BE} temperature coefficients are matched through individual collector-current adjustments,[35] but this has not been applied in production.

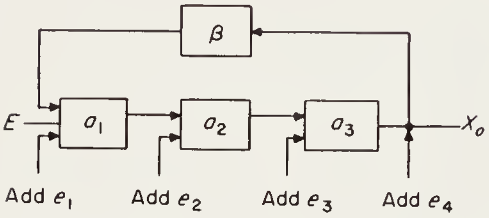


Fig. 3-31. Effect of d-c drift and noise in different amplifier stages. Since

$$X_0 = \frac{a_1a_2a_3e_1 + a_2a_3e_2 + a_3e_3 + e_4 + a_1a_2a_3E}{1 - a_1a_2a_3\beta}$$

the equivalent input offset (or noise) voltage E required to cancel the effect of the error voltages $e_1, e_2, \ldots$ is independent of feedback:

$$E = -\left(e_1 + \frac{1}{a_1}e_2 + \frac{1}{a_1a_2}e_3 + \frac{1}{a_1a_2a_3}e_4\right)$$

It follows that *low-drift/low-noise design is most important in the earlier amplifier stages.*

Current drift, which may be thought of as a change in input-transistor bias-current requirements, is due to changes in i_c/β and in the leakage current. I_{CBO} changes will, in fact, *counteract* the i_c/β change with temperature. It is possible to compensate for the current drift with the aid of temperature-dependent current sources comprising thermistors, diodes, or transistors,[37,41] or to counteract changes in β by corresponding temperature-dependent changes in i_c, but such compensation schemes work best over limited temperature ranges (perhaps 10 to 20 deg C) because of the nonlinear temperature dependence of β. In general, *a good method for reducing current drift is to use high-β transistors and to keep the first-stage collector currents below* 100 *to* 200 μA. This reduces

Table 3-3. Specifications for Computer D-c Amplifiers

1. *Rated d-c output voltage vs. current*
2. *Maximum open-loop* output voltage with specified total harmonic distortion vs. frequency* (*brief specifications* should state maximum frequency for full rated output voltage); at rated output current and at a specified fraction of rated output current
3. *Open-loop gain and phase shift vs. frequency* for small-signal (5 percent of rated max amplitude) sinusoidal output
   a. No load
   b. With specified resistance-capacitance loads (*brief specifications* should state −3-db frequency, 0-db frequency, phase shift at 0 db, and maximum phase shift with gain above 0 db)
4. *Open-loop* step response* (rise and fall times, percent overshoot) for positive and negative steps of 5 and 80 percent of max rated output amplitude
   a. No load
   b. With specified resistance-capacitance loads
5. *Maximum offset voltage* and *bias current* (referred to input) 10 min, 8 hr, and 1 week after balancing. Specify temperature range and offset voltage and current change per deg C and for specified power-supply-voltage changes
6. *RMS and peak-to-peak* noise (referred to input); give amplitude of line-frequency components separately
7. *Overload-recovery time* after specified overload condition
8. *Power-supply and environmental requirements*, *physical dimensions*, etc.
9. *Input impedance* (Z_G) *and output impedance* (Z_I)
10. *Common-mode rejection ratio* (*CMRR*) vs. frequency

* Manufacturers also often specify frequency response and step response for a unity-gain inverter with specified feedback circuit, specified resistance-capacitance load, and specified summing-point shunt capacitance, as in a computer.

i_c/β in toto, and hence also its changes with temperature. The input current can be further decreased through inclusion of grounded-collector input transistors or Darlington circuits, which roughly divide the bias-current requirement by another factor of β, but this may reduce integrated-circuit production yields. *Heated-substrate integrated circuits* have been used to keep the input-stage temperature within about 2 deg of 90 deg C, where base currents are low and less temperature-dependent.[64]

Heated-substrate transistor pairs and preamplifiers are available on monolithic chips[42] but can also be assembled from general-purpose monolithic transistor arrays.[43] A heated-substrate preamplifier may need low-impedance emitter-follower output to reduce voltage changes caused by current drift in the following stage.[42]

The heated-substrate technique has largely yielded to special integrated circuits permitting *base-current compensation* with other transistor base currents[94] or with *super-beta transistor input stages*, with β as large as 5,000 (Fig. 3-32*b*). With a collector current of 5 μA, this would reduce the *total low-temperature bias current to* $i_c/\beta = 1$ nA, and differential current drift will be of the order of 20 pA/deg C over −50 to +125 deg C (Table 3-5).[37] Voltage drift is about 5 μV/deg C.

Junction-FET input stages (Fig. 3-31*c*) have low leakage currents (of the order of picoamperes) at room temperature, as well as high input

Table 3-4. Range of Representative Computer D-c Amplifier Specifications

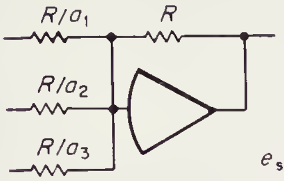
	Gain, d-c, at 25 deg C, volts/volt	Rated output		Frequency response			Input voltage drift (differential)			Input current drift (one side)			Voltage noise, d-c		
		±Volts	mA	Unity gain, MHz	Full output, KHz	Slew rate, volts/μsec	Max vs. temp., μV/deg C	Max vs. time, μV/day	Max vs. supply, μV/%	Max vs. temp., nA/deg C	Max vs. supply, nA/%	Z_{in}, open loop, megohms	To 2 Hz, μV, p-p	To 10 KHz, μV, rms	Common-mode rejection, d-c to 100 Hz, db
Bipolar-transistor input stage	10^5 10^6	11	2 500	1 50	50 5,000	5 200	10 0.8*	50 10	20 10	100 2	2 0.3	0.1 4	—	8 4	80 100
FET input stage	10^5 10^6	11	10 100	1 50	100 2,000	5 100	50 3	50 10	20 10	†	0.2 0.1	4 10	—	8 4	80 100
FET-chopper stabilized	10^6 10^8	11	10 100	1 50	100 2,000	5 100	1 0.2	1	0.5	0.05	0.5 $\times 10^{-3}$	0.3	10 5	10	—
Super-beta-transistor input stage (or heated substrate)	10^5 10^6	11	10 100	1 50	100 2,000	5 100	5‡ 0.5	2	20 10	0.05	0.2	0.4 10	—	8 4	80 100
100-volt chopper-stabilized	10^6 10^8	105	25 50	1 5	100	10 100	3 0.5	5 2	1	0.05	0.1	0.3	1,000 500	500	—

* Over small temperature range.

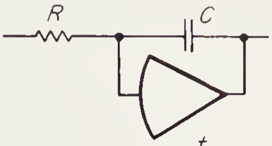
† 10 to 100 pA at 25 deg C; doubles every 10 deg C.

‡ Super-beta transistors.

Table 3-5. Typical Drift Errors in General-purpose Analog Computers Maintained at 25 ± 5 deg C, with Regulated Power Supplies*



$e_{sum} = -(a_1+a_2+a_3+1)E_d - Ri_d$



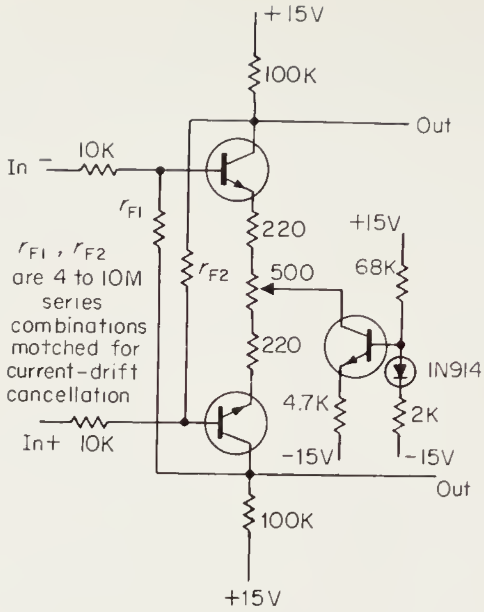
$e_{int} = -\frac{t}{RC}(E_d + Ri_d)$

	Inexpensive integrated circuit, ±10 volts, no chopper, $R = 10K$	±10-volt FET, input, no chopper, $R = 10K$ to 100K	Selected super-beta IC (or heated substrate), no chopper, $R = 10K$ to 100K	High-quality chopper-stabilized amplifier (FET chopper)	
				±10 volts $R = 10K$ to 100K	±100 volts $R = 100K$ to 1M
E_d ($\approx E_{d1} - E_{d2}$).......	±30 μV	±60 μV	±20 μV	±10 μV	±40 μV
i_d ($\approx i_{d1}$;i_{d2} not considered)..........	±30 nA	Negligible	Negligible	Negligible	Negligible
e_{sum} ($a_1 + a_2 + a_3 = 1$)	±360 μV	±120 μV	±40 μV	±20 μV	±80 μV
e_{sum} ($a_1 + a_2 + a_3 = 21$)	±960 μV	±1.3 mV	±440 μV	±220 μV	±880 μV
e_{int} (typical computer run, $t/RC = 50$)....	±16.7 mV	±3 mV	±1 mV	±500 μV	±2 mV
e_{int} (long integration, $t/RC = 1{,}000$)	±0.33 V	±60 mV	±20 mV	±10 mV	±40 mV

* See also Secs. 3-15 and 3-29.

impedances. The voltage drift of a FET d-c amplifier stage is reduced through selection of an optimal operating point;[45] matched differential FET stages are obtained by device selection or with special "interdigitated" chip layout. The best FET-input d-c amplifiers still drift about 2 μV/deg C, compared to 1 μV/deg C for good bipolar-transistor input stages. Moreover, *FET leakage currents double, approximately, for every* 10 *deg C of temperature increase*, so that super-beta or current-compensated bipolar input stages should be considered for ambient temperatures above 50 deg C (see also Fig. 3-32*d*).[37]

3-34. Operational-amplifier Stability. Bode-plot Analysis of Stability and Accuracy. Practically all one-amplifier circuits using transistor operational amplifiers obey the *restricted Nyquist-Bode law*;[39] such circuits are stable if their *phase margin* is positive, i.e., if the loop phase shift arg $(-A\beta)$ does not exceed 180 deg while the absolute loop gain $A\beta$ is greater than unity.[34] Behavior of such feedback circuits is conveniently described in terms of the well-known *Bode plot* (Fig. 3-33). For a given percentage dynamic error ϵ, the absolute loop gain $|A\beta|$ must exceed $100/\epsilon$ (Sec. 3-12); stability requires the logarithmic plot of $|A\beta|$ to cross 0 db (unity gain) at a negative slope smaller than 20


+15V
100K
Out
In −
10K
r_{F1}
r_{F1}, r_{F2} are 4 to 10M series combinations matched for current-drift cancellation
r_{F2}
220
500
220
+15V
68K
1N914
4.7K
2K
−15V
−15V
In+ 10K
Out
100K
+15V

transistors are 2N2222

Fig. 3-32*a*. Typical differential-amplifier input stage. Monolithic construction provides V_{BE} and β matching within 10 percent over the design temperature range. A current source in the emitter circuit provides high common-mode rejection with or without common-mode voltage feedback and is diode-compensated for V_{BE} changes. The crossed base-current resistors help to reduce current drift.

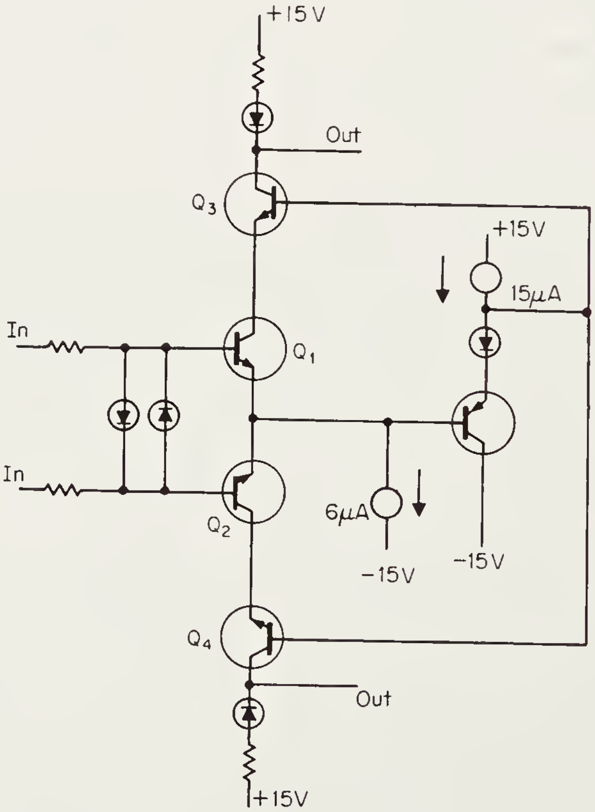


Fig. 3-32*b*. The super-beta transistors Q_1, Q_2 in this integrated-circuit input stage have low breakdown voltages (4 volts); their collector-base voltages are reduced to near zero by Q_3, Q_4. This also reduces leakage currents.[37]

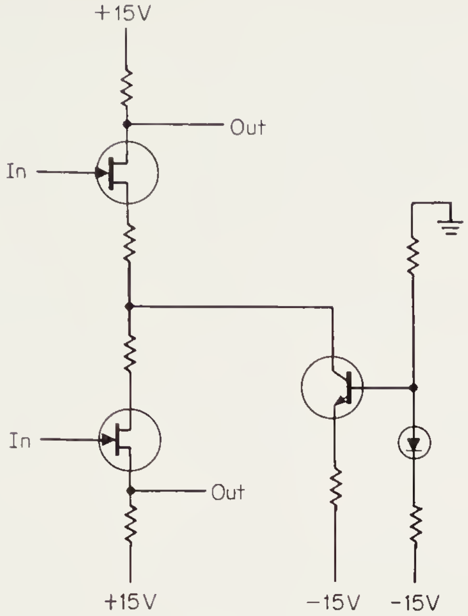


Fig. 3-32*c*. Junction FET input stage.[45]

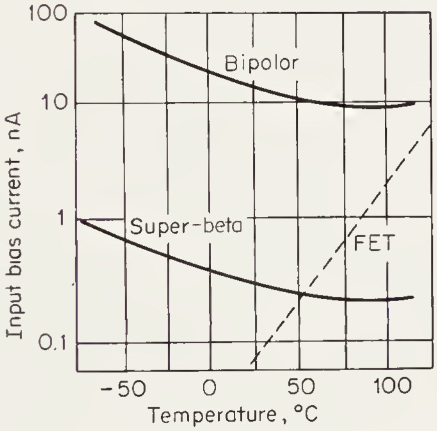


Fig. 3-32*d*. D-c amplifier input bias current vs. temperature. Differential offset currents would be smaller by about a factor of 10.

db/decade.[34] The forward-gain slope of many general-purpose amplifiers never exceeds 20 db/octave over the entire frequency range, but special-purpose amplifiers may be given a 40 db/octave hump near the working-frequency region for extra gain (see also Sec. 3-18*b*). *Amplifier stability or phase margin must be specified with stated capacitive loads on output and summing point.* If the phase margin is insufficient, it will be necessary to employ equalization (*rolloff networks*) to decrease the loop gain over a portion of the frequency range (Fig. 3-33). Most integrated-circuit amplifiers have terminals for connecting such equalization networks, as detailed on their specification sheets. Note that capacitive rolloff networks may take substantial current and will therefore tend to

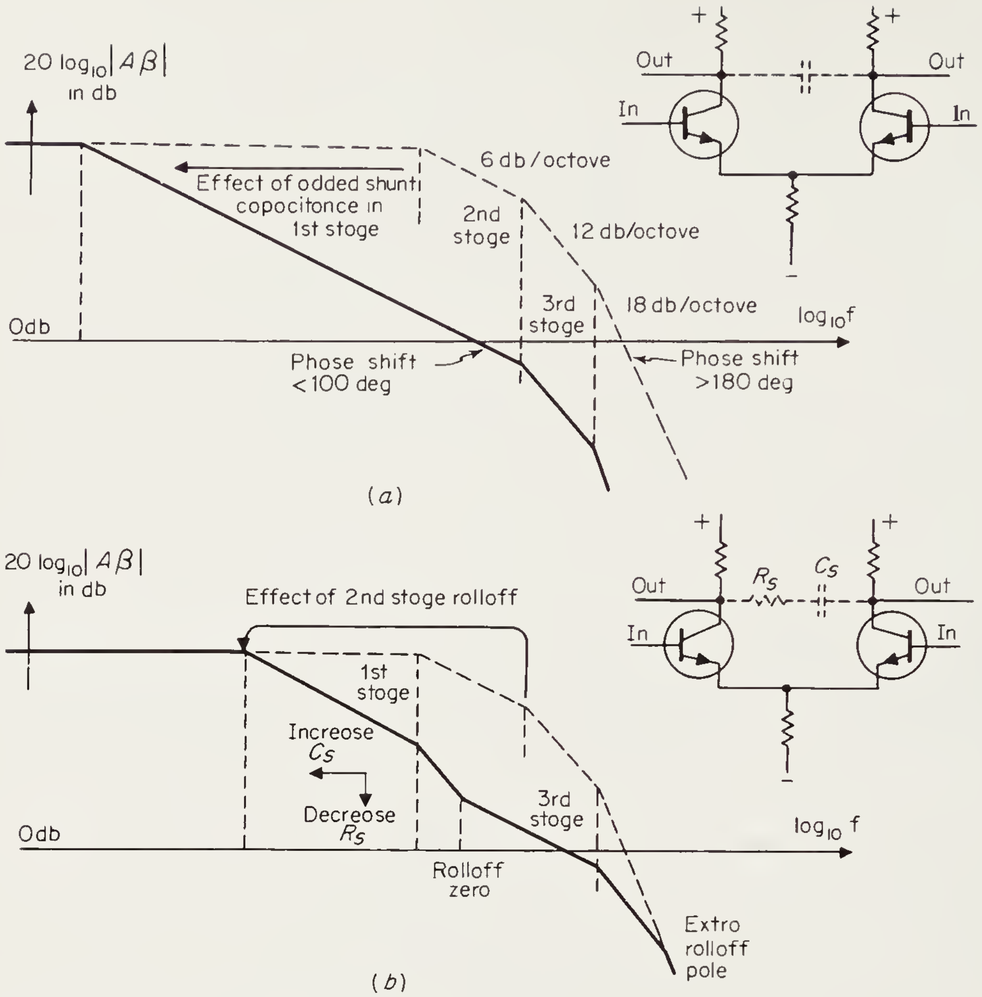


Fig. 3-33. Bode plot of $20 \log_{10} |A(j\omega)\beta(j\omega)|$ vs. $\log_{10} f = \log_{10} (\omega/2\pi)$ for a multistage amplifier, and two schemes achieving stability at the expense of some high-frequency response. The compensated plots cross the unity-gain (0-db) line at −6 db/octave. Many integrated-circuit amplifiers have special terminations for external rolloff networks. Design must allow for load and circuit capacitances (see Refs. 34 and 39).

reduce power bandwidth and slewing rate. For this reason, one tries to apply rolloff networks to low-level stages, whenever practical. Figure 3-34 shows a number of circuit arrangements for rolling off existing amplifiers without such rolloff terminals.

3-35. Feedforward Circuits. Feedforward amplifiers (Fig. 3-35) pass high-frequency signal components through a high-pass channel directly to a wideband output stage, while cascading a high-gain d-c amplifier with the output stage for lower frequencies. The feedforward channel may or may not include amplification.

±10-volt class AB complementary-transistor output stages, like that in Fig. 3-35, have low quiescent current and can have gains as high as 2,000 at 200 KHz, with unity gain at 20 to 100 MHz, depending on transistors and load. In Fig. 3-35, output-driver gain has been traded

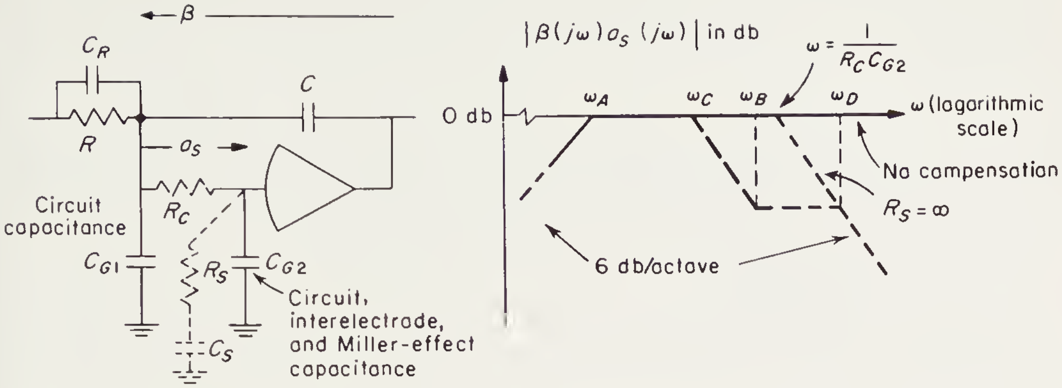


Fig. 3-34. Feedback- and compensating-network frequency response for an integrator (see also Sec. 3-20). Assuming $C_{G1} + C_{G2} + C_R \ll C$, we have

$$\beta(s) = \frac{s}{s + \omega_A} \qquad \text{with } \omega_A = \frac{1}{RC}$$

If the compensating network (R_C alone, or R_C, R_S, and C_S) is used, then at low frequencies, $\beta(s)a_S(s) \approx \beta(s)$, where the compensating-network response $a_S(s)$ counts as part of the amplifier forward gain $A(s)$. At frequencies high compared with $\omega_A/2\pi$,

$$\beta(s)a_S(s) \approx a_S(s) = \frac{\omega_C\omega_D}{\omega_B}\,\frac{s + \omega_B}{(s + \omega_C)(s + \omega_D)}$$

with $\omega_B = 1/R_SC_S$, $\omega_C \approx 1/R_CC_S$, $\omega_D \approx 1/R_SC_{G2}$ if we can assume $R_S \ll R_C$, $C_S \gg C_{G2}$.

for low open-loop output impedance by local degeneration. Such series-transistor circuits are especially well suited to feedforward injection, since high- and low-frequency signals are added without any mixing loss. The combination of such a feedforward output stage with a modern low-cost integrated-circuit input amplifier yields a number of remarkable advantages:[56–69]

1. *The amplifier high-frequency response is essentially that of a single wideband stage followed by an emitter follower*, and no shunt rolloff networks are required for stabilization. As a result, such amplifiers can produce full ±10 volts, 30 mA output all the way up to 5 MHz.
2. *The feedforward combination completely avoids the usual compromise between high-frequency response and low-drift/low-noise design of the d-c amplifier input stage.* If this fact were more generally realized, the current drift and low-frequency-flicker noise of many general-purpose integrated-circuit d-c amplifiers could be improved by at least a factor of 3 by the simple expedient of reducing their input-stage collector current; the feedforward connection would take care of the frequency response.

One can improve the small-signal *and* power bandwidths of many existing integrated-circuit amplifiers by introducing feedforward into

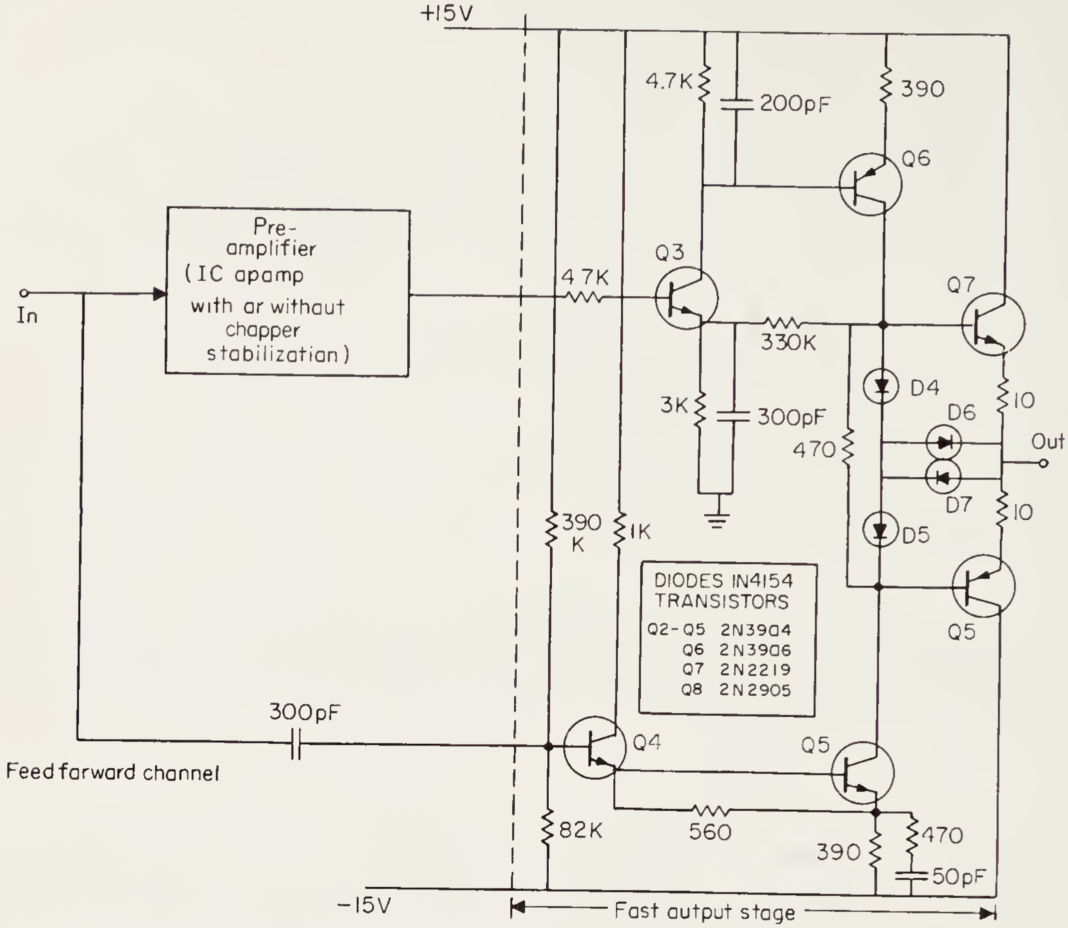


Fig. 3-35. Design of a modern high-quality ±10-volt amplifier. The powerful class AB output stage has diode overcurrent protection and 40-db gain at 300 KHz, 0 db at 30 MHz. A feedback-pair driver gives the amplifier very low open-loop output impedance (100 ohms at 100 Hz, 50 ohms at 10 KHz). Low- and mid-frequency gain (120 db at d-c with 20 db/octave rolloff) are provided by an inexpensive integrated-circuit operational amplifier with or without chopper stabilization. Feedforward around this preamplifier produces a unity-gain bandwidth of 30 MHz, with full ±10-volt 30-mA output to 5 MHz. (*University of Arizona; see also Refs.* 64 *and* 65.)

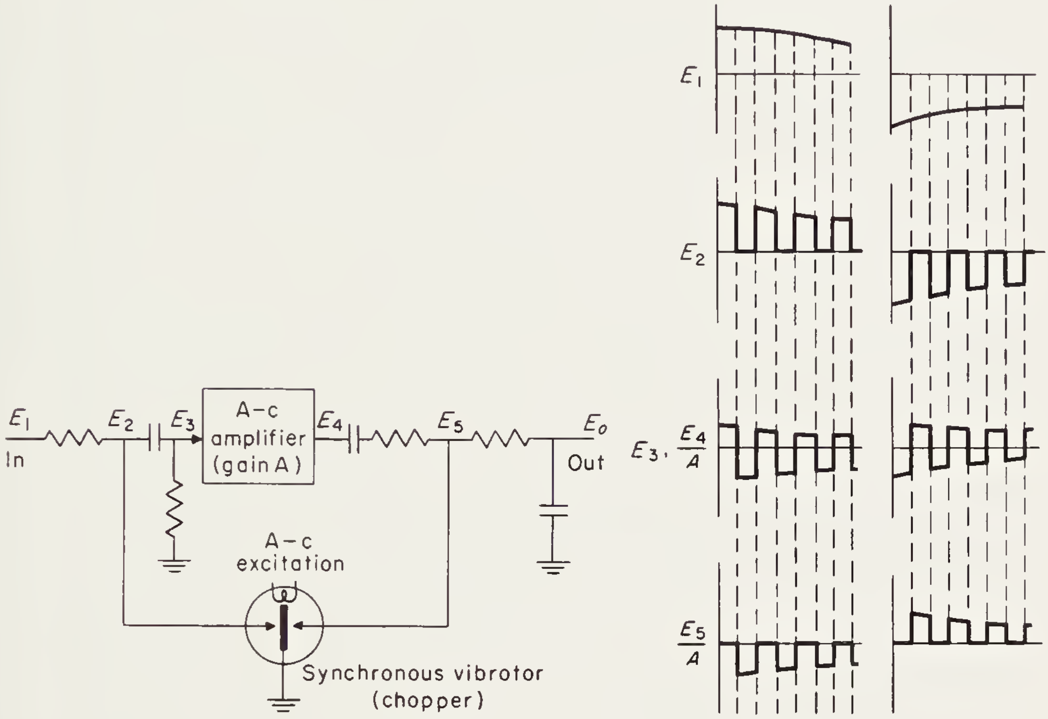
later-stage rolloff terminals; rolloff equalization may or may not be needed.

Under very general conditions, proper feedforward-circuit design requires that the low-pass-preamplifier phase shift be less than 180 deg where the preamplifier gain exceeds unity (see also Sec. 3-34).[59] Otherwise, there will be serious phase shift and attenuation near the preamplifier unity-gain frequency. This manifests itself in the *transient response* of many feedforward amplifiers (oscillations or long settling times determined by the preamplifier unity-gain bandwidth).

Differential feedforward amplifiers exist but are relatively rare.[39]

3-36. Chopper-stabilized D-c Amplifiers. (a) To obtain the very lowest voltage and current drift for d-c amplification, one employs a-c-operated relay or electronic chopper switches, photoresistors, or varactor diodes to modulate the d-c amplifier input signal. After drift-free a-c amplification, phase-sensitive demodulation by a second chopper followed by a low-pass filter recovers the amplified d-c signal (Fig. 3-36*a*, *b*).

Figure 3-36*c* shows a feedforward scheme combining such a low-offset carrier-type amplifier (*chopper-stabilizer channel*) with a conventional d-c amplifier to improve its drift and low-frequency gain without any bandwidth reduction; the d-c amplifier may well include additional high-frequency feedforward in the manner of Fig. 3-35. In terms of the analysis given in Fig. 3-31, the chopper-stabilizer channel in Fig. 3-36*c* acts as a low-frequency preamplifier, so that *drift originating in the main amplifier is effectively divided by the absolute d-c gain* $|A_2|$ *of the stabilizer channel*, typically 1,000 to 4,000. As a result, practically the entire remaining drift of the chopper-stabilized d-c amplifier is due to:



(*a*) (*b*)

Fig. 3-36*a*, *b*. Carrier-type or modulator/demodulator d-c amplifier using a single electromechanical chopper (*a*), and waveforms (*b*).

1. Chopper-frequency pickup, including ground currents, in the stabilizer channel
2. Small offset voltages originating in the chopper itself
3. Leakage into summing-point or stabilizer-input connections
4. D-c offset voltages in the ground circuits

In typical computer installations, the remaining long-term (8-hr) offsets are within 3 to 100 μV, 0.2 μV/deg C, and 30 pA/deg C over a wide temperature range. Even smaller offsets can be obtained in instrument-type amplifiers.

In addition to the drift reduction afforded by the stabilizer channel, the extreme-low-frequency gain of the main amplifier is multiplied by

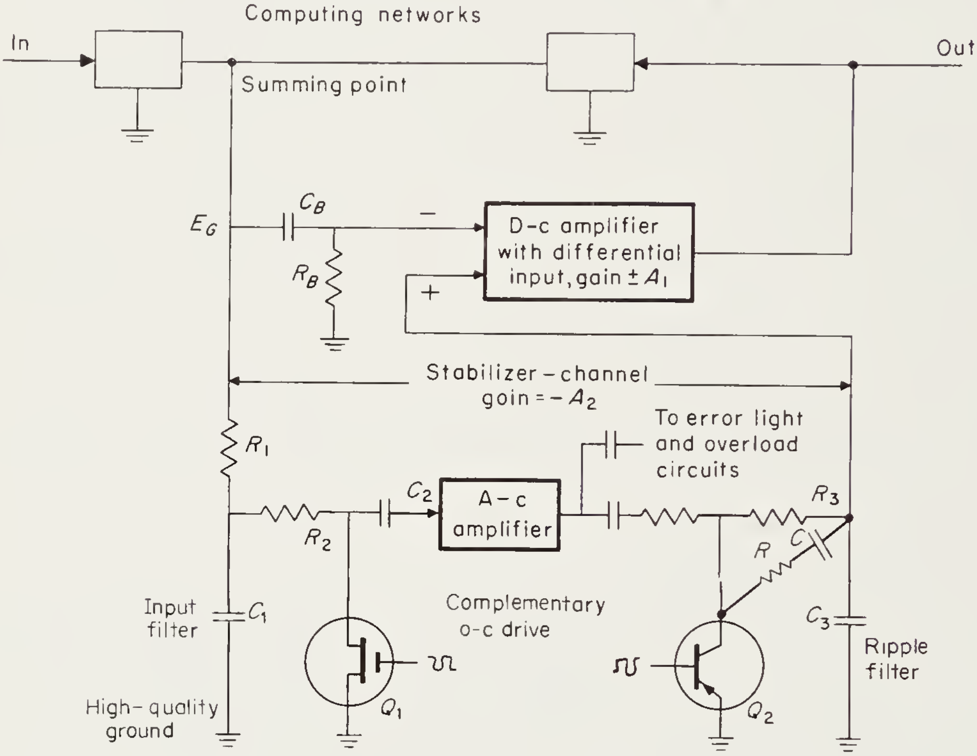


Fig. 3-36c. D-c amplifier with chopper stabilization. MOSFET Q_1 is ON when transistor Q_2 is OFF, and vice versa, to produce the inverted chopper-channel output, as in Fig. 3-36*a*, *b*. Capacitors must have low dielectric absorption (Mylar or polystyrene). It is best to include a fast overload limiter (Fig. 4-8*d*) to prevent blocking of the chopper channel with its relatively long time constants. $C_{B'}$, R_B can be omitted if the main d-c amplifier has low input current (FET input). The increased summing-point voltage due to even a mild overload produces a greatly increased a-c output in the stabilizer channel. The amplifier pulses at the a-c amplifier output furnish a very useful indication of voltage *and* current overload (*error lights*).

$|A_2|$; thus, for $A_1 = 50{,}000$, $|A_2| = 2{,}000$, we obtain the phenomenal d-c gain $|A| = |A_1A_2| = 100{,}000{,}000$ combined with low drift, so that excellent low-frequency computing accuracy becomes possible.

The design of bipolar and FET switches for chopper applications is discussed in Chap. 5. MOSFET choppers operated at frequencies of 100 to 1,000 Hz have long-term input-voltage offsets below 20 μV and negligible current offset over a wide range of temperatures. Sinusoidal drive to the input chopper reduces switching spikes. FET choppers permit low-voltage drive at higher frequencies than neon-bulb/photoresistor choppers and are also more reliable. With modern low-drift amplifiers (Sec. 3-33), chopper stabilization is, however, required only in the most accurate computer and instrumentation circuits.

(b) Stabilizer-channel Filters and Frequency Response. The stabilizer input and output filters (R_1, C_1 and R_3, C_3 in Fig. 3-36*c*) cut off well below the chopper frequency f_C, and dominate the stabilizer-channel frequency response, so that for practical purposes

$$A_2(j\omega) = \frac{A_2(0)}{(j\omega R_1 C_1 + 1)(j\omega R_3 C_3 + 1)}$$

The *input filter* (R_1,C_1) attenuates chopper-frequency input noise and prevents stabilizer-channel errors or overloads due to the possibly relatively large high-frequency components of the summing-point error voltage E_G. C_1 and R_2 also form a filter intended to remove chopper-frequency components from the amplifier summing point. R_1C_1 varies between $0.8/f_C$ and $10/f_C$, with larger values corresponding to larger stabilizer-channel gains and larger expected summing-point errors.

The *ripple filter* (R_3,C_3) smoothes the stabilizer output ripple, which acts as a noise voltage (chopper noise) at the amplifier input. Since the d-c output of our stabilizer must approximate the d-c offset E_{DC} of the main d-c amplifier in absolute value, the amplitude of the rectified output-chopper pulses is about $2\,|E_{DC}|$, assuming symmetrical chopper action. The ripple filter reduces the ripple to a roughly triangular waveform at the chopper frequency f_{CHOPPER}, with peak-to-peak amplitude $E_{DC}/R_3C_3f_C$. With a d-c offset $E_{DC} = 12$ mV, for instance, a filter time constant $R_3C_3 = 2$ sec in Fig. 3-36*c* will reduce 600-Hz chopper noise to 10 μV peak to peak. Note that balancing of the main d-c amplifier will reduce E_{DC}, and hence the error voltage in the chopper channel and the chopper noise.

An excessively large filter time constant R_3C_3 adds to amplifier cost, restricts the benefits of the chopper-channel gain, and slows overload recovery. To reduce chopper noise without increasing R_3C_3, one can:

1. Increase the chopper frequency
2. Use a more complicated chopper circuit, which breaks the connection to R_3 before grounding the a-c amplifier output to obtain peak detection or sample-hold action
3. Add an a-c preamplifier ahead of the main d-c amplifier to divide the chopper noise by the preamplifier gain

In practice, however, the feedback-stability requirements outlined in Sec. 3-34 may require R_3C_3 to be larger than needed for adequate ripple filtering, viz.,

$$R_3C_3 > 2A_2(0)R_1C_1 \text{ to } 4A_2(0)R_1C_1 \tag{3-96}$$

This condition ensures that the output filter reduces the stabilizer-channel gain $|A_2(j\omega)|$ to unity before the input filter causes excessive phase shift; $|A_2(j\omega)|$ should also reduce to unity before the main-amplifier phase shift exceeds, say, 45 deg. Where the filter time constant R_3C_3 required by Eq. (96) seems excessive, we can also "break" the stabilizer-channel gain with a rolloff network (R, C in Fig. 3-36*c*).[49]

3-37. Balancing and Test Circuits. *Voltage balancing* to minimize the offset voltage $E_d \approx E_{d1} - E_{d2}$ may be done in the amplifier input stage through differential adjustments of the input-transistor currents (emitter resistors or collector voltages in Fig. 3-32). Figure 3-37 shows an external voltage-balancing circuit, which can be added to existing amplifiers. Voltage balancing may be done in a test circuit like that of Fig. 3-38*a*.

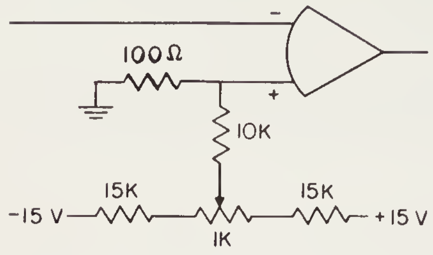
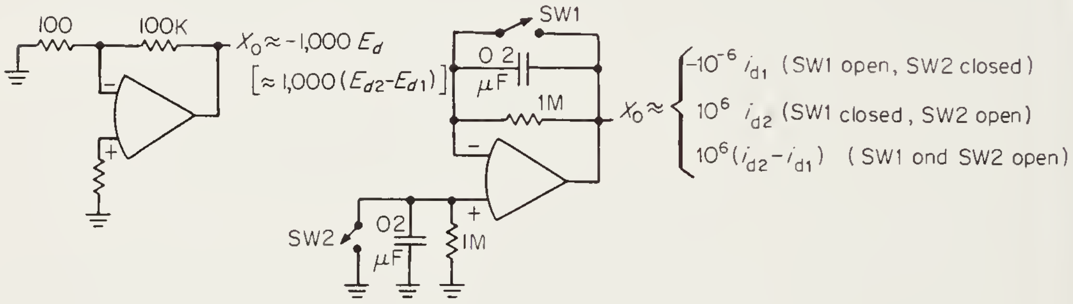


Fig. 3-37. This external voltage-balancing circuit can be added to existing amplifiers.

Current balancing of bipolar-transistor-input d-c amplifiers to minimize i_{d1} and i_{d2} in a test circuit similar to Fig. 3-38*b* is usually performed during manufacture through adjustment of the base-current bias resistors in Fig. 3-32*a*.

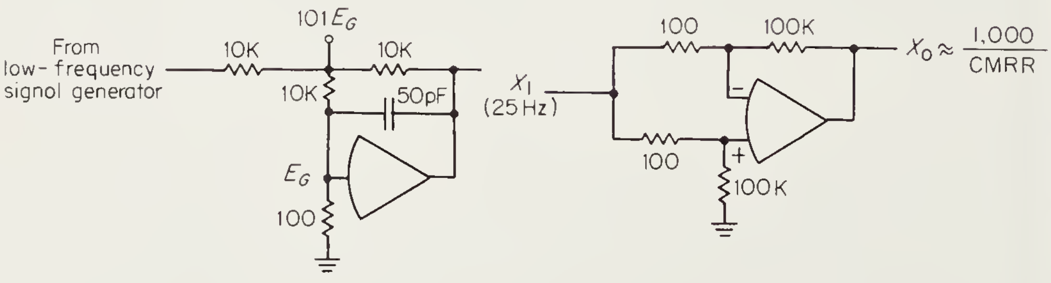
Figure 3-38 shows some *circuits used to test amplifier specifications.* Additional test circuits and, in particular, circuits for measuring amplifier input and output impedances will be found in Refs. 38, 39, 46, and 74. The reader is advised to consult manufacturers' manuals for the specific test circuits used to define catalog specifications.



100
100K
-
+
$X_0 \approx -1{,}000\,E_d$
$[\approx 1{,}000(E_{d2}-E_{d1})]$
SW1
0.2 μF
1M
-
+
SW2
0.2 μF
1M
$X_0 \approx$
$-10^{-6}\,i_{d1}$ (SW1 open, SW2 closed)
$10^{6}\,i_{d2}$ (SW1 closed, SW2 open)
$10^{6}(i_{d2}-i_{d1})$ (SW1 and SW2 open)

(*a*) (*b*)

Fig. 3-38*a* and *b*. (*a*) Measurement of input offset voltage. (*b*) Measurement of input bias and offset currents.



From low-frequency signal generator
10K
$101E_G$
10K
10K
50pF
E_G
100
X_1 (25 Hz)
100
100K
-
+
100
100K
$X_0 \approx \frac{1{,}000}{CMRR}$

(*c*) (*d*)

Fig. 3-38*c* and *d*. (*c*) Measurement of summing-point voltage for error and open-loop-gain estimation at frequencies below 1 KHz. Careful shielding may be necessary (Ref. 74; see also Sec. 3-38). (*d*) Measurement of the common-mode-rejection ratio (*CMRR*).

3-38. Practical Frequency-response Measurements. The tremendous d-c gain of modern chopper-stabilized d-c amplifiers makes direct measurement of the zero-frequency gain practically impossible. Quoted d-c gain figures are either calculated theoretically or obtained from separate gain measurements in the d-c amplifier and chopper-stabilizer channels. Fortunately, however, the frequency-response (gain and phase) measurements most important to the design process are made at higher frequencies, say in excess of 1 KHz, where the amplifier gain has more manageable values. Two practical circuits for frequency-response measurements with simple equipment are suggested:

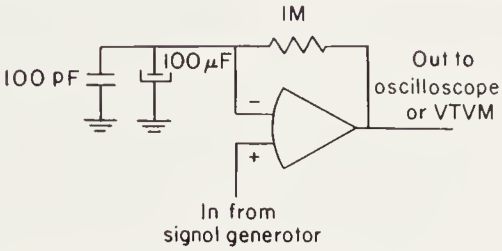


Fig. 3-39. Circuit for measuring the open-loop performance of a differential-input d-c amplifier above 1 KHz. Similar connections are useful when such a d-c amplifier is to serve as an a-c amplifier.

1. Connect the amplifier under test as a unity phase inverter with sinusoidal input. Vary the input to obtain a standard undistorted output amplitude, say 10 volts for ±100-volt amplifiers, and measure the amplitude at the summing point with a vacuum-tube voltmeter and/or oscilloscope. Input and output phases may be compared with the aid of a dual-trace oscilloscope.
2. Attenuate the amplifier low-frequency response with the feedback networks shown in Fig. 3-39. Disconnect the chopper stabilizer and apply a known sinusoidal signal to the chopper-stabilizer (difference-amplifier) input as shown. Measure input and output.

The second method also permits one to measure open-loop output distortion at the higher frequencies, and hence the maximum output amplitude vs. frequency for specified total harmonic distortion.

While phase comparisons of two signals having a large amplitude ratio are relatively inaccurate with the simple equipment described, the amplitude response can be conveniently read directly in decibels with a suitably calibrated, well-shielded signal generator and a logarithmic voltmeter. One can enter the amplitude response directly into a Bode plot and draw straight-line asymptotes to determine frequency breaks (Sec. 3-34). It is then often reasonable to omit explicit phase measurements altogether.

If a differential-input oscilloscope is available, one can measure the phase difference ϑ between two sinusoidal signals $a \sin \omega t$ and $b \sin (\omega t + \vartheta)$ as follows. Display the difference signal

$$a \sin \omega t - b \sin (\omega t + \vartheta) = r \sin (\omega t + \varphi) \tag{3-97}$$

with

$$r = \sqrt{a^2 + b^2 - 2ab \cos \vartheta} \tag{3-98}$$

Vary the oscilloscope gain controlling b so as to minimize the difference-signal amplitude (98); this results in

$$b = a \cos \vartheta \qquad r = a \sin \vartheta$$

so that $|\sin \vartheta|$, $|\cos \vartheta|$, or $|\tan \vartheta|$ can be obtained as an amplitude ratio. The correct quadrant of the unknown phase difference ϑ is determined by observation of the two signals, unless it is known from the nature of the circuit being studied.

REFERENCES AND BIBLIOGRAPHY

1. Miller, K. S., and F. J. Murray: The Mathematical Basis for the Error Analysis of Differential Analyzers (2 parts), *J. Math. Phys.*, nos. 2 and 3, 1953.
2. Meissinger, H. F.: Substitution Methods for the Verification of Analog Solutions, *Proc. Natl. Simulation Conf.*, Dallas, Tex., 1956, Simulation Councils, Inc., La Jolla, Calif.; reprinted in *Simulation*, December 1964.
3. Tomovič, R., and W. J. Karplus: *High-speed Analog Computers*, Wiley, New York, 1962.
4. Korn, G. A.: Parameter-perturbation Generator for Analog-computer Error and Optimization Studies, *Ann. AICA*, April 1963.

Potentiometers

5. Nettleton, L. A., and F. E. Dole: Potentiometers, *Rev. Sci. Instr.*, **17**: 356 (1946).
6. Loveman, B. D.: Coefficient-setting Potentiometers, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
7. Gilbert, J.: Shortcut in Compensating Potentiometer Loading Errors, *Control Eng.*, February 1955.
8. ———: Compensating Functions Potentiometers for Loading Errors, *Control Eng.*, March 1955.
9. Single, C. H., and J. A. Brussolo: Copper-mandrel Potentiometer Dynamic Error and Compensation, *IRETEC*, September 1961.

Resistors and Capacitors

10. Single, C. H.: Precision Components for Analog Computers, *ISA Convention Record*, paper 56-21-2, September 1956.
11. Sykes, R. P.: Calibration of High-resistance Padded Resistors, *Instr. Control Systems*, October 1959.
12. ———: Calibrating Analog-computer Integrators, *Instr. Control Systems*, September 1960.
13. Single, C. H., and E. M. Billinghurst: Optimization of Analog-computer Linear System Dynamic Characteristics, *Proc. Western Joint Computer Conf.*, Los Angeles, Calif., May 1961.
14. ———: Linear-element Error Sources and Effects, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
15. Dow, P. C.: Capacitor Dielectric Absorption, *IRETEC*, March 1958.
16. Miura, T., et al.: Some Considerations of the Error Due to Absorption Phenomena in Integrating Capacitors, *Proc. 2d AICA Conf.*, Strasbourg, France, 1958, Presses Académiques Européennes, Brussels.
17. Brussolo, J., and C. H. Single: Transient Analysis of Capacitor Equivalent Circuit, *Beckman/Berkeley Eng. Rept.* CRD 59-5, 1959.
18. Spirer, H. F., and S. Rudnick: Precision Resistors, *Electromechan. Design*, September 1965.
19. Smith, P. D.: High-frequency Characteristics of Precision Film Resistors, *Application Note*, Resistor Department, Texas Instruments, Inc., Dallas, Tex., 1960.
20. Johnston, S. A.: Selecting Trimmer Potentiometers for High-frequency and Pulse Applications, *Application Note*, Amphenol Controls Division, Janesville, Wis., 1968.
21. Single, C. H., and T. A. King: Measurement of Resistor Dynamic Characteristics, *Simulation*, October 1968.

Compensation of Capacitor Dielectric Absorption

22. Sheehan, C. J.: An analog-computer Simulation of the Restricted Three-body Problem, *IEEETEC*, February 1964.
23. Single, C. H.: *U.S. Patent Application*, assigned to Applied Dynamics Corp., Ann Arbor, Mich., 1965.
24. Guyton, R. D.: A Passive-network Method of Compensating an Analog Integrator, *Simulation*, February 1968.

Errors in Linear-system Solutions

25. Korn, G. A.: Exact Design Equations for Operational Amplifiers with Four-terminal Computing Networks, *IRETEC*, February 1962.
26. Miura, T., et al.: On Computing Errors of an Integrator, *Proc. 2d AICA Conf.*, Strasbourg, France, 1958, Presses Académiques Européennes, Brussels.
27. MacNee, A. B.: Some Limitations on the Accuracy of Electronic Differential Analyzers, *Proc. IRE*, **40**: 303 (March 1950).
28. Marsocci, V. A.: An Error Analysis of Electronic Analog Computers, *IRETEC*, December 1956.
29. Dow, P. C.: An Analysis of Certain Errors in Electronic Differential Analyzers (2 parts), *IRETEC*, December 1957, March 1958.
30. Miura, T., and M. Nagata: Theoretical Considerations of Computing Errors of a Slow-type Electronic Analog Computer, *IRETEC*, December 1958.
31. Sosenskii, N. L.: On the Determination of Certain Errors in Analog Computers, *Automatika i Telemekhanika*, Oct. 20, 1959.
32. Kogan, B. Y.: On the Steady-state Operational-amplifier Errors Due to the Passband Limitations, *Ann. AICA*, Apr. 4, 1962.
33. Brubaker, T. A.: Design, Development, and Applications of ASTRAC I, Ph.D. Thesis, Electrical Engineering Department, University of Arizona, 1963; see also *Ann. AICA*, April 1964.

D-c Amplifiers: General

34. Thornton, R. D., et al.: *Multistage Transistor Circuits*, Wiley, New York, 1965.
35. Hoffait, A. H., and R. D. Thornton: Limitations of Transistor d-c Amplifiers, *Proc. IEEE*, February 1964.
36. Meyer-Brötz, G., and E. Heim: Ein breitbandiger Operationsverstarker mit Silizium-Transistoren, *Telefunken Z.*, January 1966.
37. Widlar, R. J.: IC Op Amp Beats FETs on Input Current, *EEE*, December 1969; *Application Note* AN-29, National Semiconductor Corporation, Santa Clara, Calif., 1969.
38. *Applications Manual*, Philbrick/Nexus Research, Dedham, Mass., 1966.
39. *Catalog and Handbook*, Burr-Brown Research Corporation, Tucson, Arizona.
40. Giles, J. N.: *Linear-integrated-circuits Applications Handbook*, Fairchild Semiconductor, Mountain View, Calif., 1967.
41. Six Bias-current Compensation Techniques, *Electronics*, Feb. 5, 1968.
42. Operational Amplifiers: Make or Buy? *Application Note*, Fairchild Semiconductor, Mountain View, Calif., 1967.
43. De Angelis, D. P., and M. Palumbo: Regulator Holds Temperature of Chip Substrate Constant, *Electronics*, Feb. 2, 1970.
44. Solomon, J. E., and W. R. Davis: A Self-compensated Operational Amplifier with Low Input Current and High Slew Rate, *Proc. Intern. Solid-State Circuits Conf.*, 1969.
45. Freyling, N.: FET Differential Amplifier, *Application Note* AN-231, Motorola Semiconductor Products, Inc., 1966.
46. Stata, R.: Operational Amplifiers, *Electromechan. Design*, September and November 1965; see also *Application Notes*, Analog Devices, Inc., Cambridge, Mass.

Chopper Stabilization

(See also references in Chap. 5 on Electronic Switching.)

47. Williams, A. J., R. E. Tarpley, and W. R. Clark: *Trans. AIEE*, **67**: 47 (1948).
48. Goldberg, E. A.: Stabilization of D-c Amplifiers, *RCA Rev.*, **11**: 206 (1950). See also

U.S. Patents 2,685,000 (A. W. Vance) and 2,684,999 (E. A. Goldberg and J. Lehmann), both assigned to Radio Corporation of America.

49. Billinghurst, E. M.: Design of Vacuum-tube D-c Amplifiers, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
50. Ehlers, H. L., and W. Hochwald: Design of a High-quality Transistor D-c Amplifier, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
51. Hamer, H.: A Stabilized Driftless Analog Integrator, *IRETEC*, June 1954.
52. Ingerson, W. E.: Drift Compensation in D-c Amplifiers, *IRE Natl. Convention Record*, 1951.
53. Schwartz, J., and R. Solomonoff: Photoelectric Chopper for Guided Missiles, *Electronics*, November 1954.
54. *Photocom Data Sheet*, James Electronics, Inc., Chicago, Ill., 1962.
55. McDonald, R. K.: Optoelectronic Components, *Electronic Inds.*, May 1963.

Feedforward Techniques

56. Deering, C. S.: A Wide-band Direct-coupled Operational Amplifier, *Proc. Natl. Simulation Conf.*, Dallas, Tex., 1956, Simulation Councils, Inc., La Jolla, Calif.
57. Koerner, H.: How to Extend Operational-amplifier Response, *Electronics*, Nov. 11, 1960.
58. Polonikov, D. E.: Wide-band Operational Amplifiers, *Automatika i Telemekhanika*, December 1960.
59. ———: *Electronic Amplifiers*, Macmillan, New York, 1965.
60. Ozdes, D.: Design of a Wideband Transistor D-c Amplifier (M.S. Thesis), *ACL Memo.* 79, Electrical Engineering Department, University of Arizona, 1963.
61. Waldhauer, F. D.: Latest Approach to Integrated-amplifier Design, *Electronics*, May 31, 1963.
62. Meyer-Brötz, G.: Bandbreitenvergleich von stark gegenkoppelbaren Operationsverstärkern, *Arch. Elektr. Übertr.*, **18:** 51 (1964).
63. Korn, G. A.: Progress of Analog/Hybrid Computation, *Proc. IEEE*, *December* 1966.
64. Naylor, J. R.: A New High-performance Computer D-c Amplifier, *Simulation*, May 1967.
65. Conant, B. K.: A New Solid-state Iterative Differential Analyzer, *Proc. FJCC*, 1968.
66. Dobbin, R.: Feedforward Compensation Speeds Op Amp, *Application Note* LB-2, National Semiconductor Corporation, Santa Clara, Calif., 1969.
67. Kley, A., and G. Meyer-Brötz: Zur Verbesserung der Bandbreite, *Archiv elektr. Übertr.*, October 1965.
68. Bladowski, R., et al.: Ein Breitbandiger Operationsverstärker mit Vorwärtskoppelung, *NTZ*, September 1967.
69. Welling, B.: Add-on Network Improves Op-amp Bandwidth and Slew Rate, *The Electronic Engineer*, January 1970.

Measurements

(See also Refs. 38, 39, 46, and 81.)

70. Handler, H.: A Technique for Measuring the Phase Margin of an Operational Amplifier, *IRETEC*, August 1962.
71. ———: Measurement of Phase Shift, *IEEETEC*, June 1963.
72. Hirschfeld, R. A.: Go/no-go Technique Tests Amplifiers, *Electron. Design*, May 10, 1967.
73. Stata, R.: User's Guide to Applying and Measuring Operational-amplifier Specifications, *Analog Dialogue*, Analog Devices, Inc., Cambridge, Mass., September 1967.

Amplifier Common-mode Rejection and Instrumentation Amplifiers

74. Offner, F. F.: Balanced Amplifiers, *Proc. IRE*, **35**:306 (1947).
75. Andrew, A. M.: Differential-amplifier Design, *Wireless Eng.*, **33:** 73 (1955).
76. Birt, D. R.: Self-balancing Push-pull Circuits, *Wireless World*, **66:** 223, 283 (1960).
77. Meyer-Brötz, G.: Grundschaltungen von Gleichspannungsverstärkern in der Messwertverarbeitung, *Intern. Elektron. Rundschau*, July 1963.

78. ———: Zum Problem der Gleichtaktunterdrückung, *Nachrtech. Z.*, February 1966.
79. ——— and A. Kley: The Design of Differential D-c Amplifiers with High Common-mode Rejection, *Electron. Eng.*, February 1966.
80. Middlebrook, R. D.: *Differential Amplifiers*, Wiley, New York, 1963.
81. Tobey, G. E., et al.: *Operational Amplifiers—Design and Application*, McGraw-Hill, New York, 1971.
82. Moss, R. Y.: Errors in Data-amplifier Systems, *Hewlett-Packard J.*, Hewlett-Packard Company, Palo Alto, Calif., July 1967.
83. Levin, M. H.: Advantages of Direct-coupled Differential Data Amplifiers, *Hewlett-Packard J.*, Hewlett-Packard Company, Palo Alto, Calif., July 1967.
84. New Operational Amplifier for Differential Instrumentation (Philbrick/Nexus), *Electron. Capabilities*, Summer 1967.
85. Holbrook, J. G., and T. H. Wilmshurst: Eliminate Common-mode Response in Closed-loop Systems, *The Electronic Engineer*, August 1968.
86. Demrow, R. I.: Narrowing the Margin of Error, *Electronics*, April 15, 1968.
87. ———: Protecting Data from the Ground Up, *Electronics*, April 29, 1968.
88. Differential Amplification, *Application Note* ZA-102, Zeltex, Inc., Concord, Calif., 1968.
89. Choosing an Amplifier for Instrumentation, *Application Note* PDS-207B, Burr-Brown Research Corp., Tucson, Ariz., February 1969.
90. Miller, W.: Eight Ways to Use Op Amps in Bridge Circuits, *Control Eng.*, February 1969.
91. Gans, F.: Common-mode Rejection Ratio: What the Spec Sheet Doesn't Say, *Electronics*, June 23, 1969.
92. Smith, L., and D. H. Sheingold: Noise and Operational Amplifier Circuits, *Analog Dialogue*, Analog Devices, Inc., Cambridge, Mass., March 1969.

Active Filters

(See also bibliography for Chap. 1.)

93. Huelsman, L.: *Theory and Design of Active RC Circuits*, McGraw-Hill, New York, 1968.

Miscellaneous

94. Graeme,: U.S. Patent 3,551,832 assigned to Burr-Brown Research Corporation, Tucson, Ariz., 1970.
95. Giacoletto, L. J.: *Differential Amplifiers*, Wiley, New York, 1970.
96. Single, C. H.: *Integrator Error Sources and Compensation Techniques*, privately published; contact Mr. Single at Applied Dynamics, Inc., Ann Arbor, Mich., 1970.
97. Morrison, R.: *D–c Amplifiers in Instrumentation*, Wiley, New York, 1970.

CHAPTER 4

NONLINEAR OPERATIONS: LIMITERS, COMPARATORS, AND FUNCTION GENERATORS

INTRODUCTION

4-1. Function Generation and Switching. This chapter describes the basic devices required to vary a voltage X_0 as a nonlinear function $F(X_1,X_2, \ldots)$ of other voltages $X_1, X_2, \ldots$. Many circuits employing nonlinear resistors, semiconductors, or vacuum tubes can produce continuous nonlinear relations $X_0 = F(X_1)$ (Secs. 4-16 and 4-17), but their characteristics are not easily reproduced with precision. For this reason, most electronic function generators employ biased diodes (rectifiers) as voltage-sensitive *switches* which select network-gain values set by accurate linear resistances. Changes in diode characteristics have, then, only a relatively small effect on the output (Secs. 4-2 to 4-7 and 4-15). The *precision limiter and selector circuits* of Secs. 4-10 and 4-11 employ operational-amplifier feedback for greatly improved accuracy. Sections 4-12 and 4-13 describe useful *signal-generator circuits. Generation of functions of two or more variables* is discussed in Secs. 4-18 to 4-21.

An increasing number of computer applications involve relations $X_0 = F(X_1,X_2, \ldots)$ where an input X_i or the output X_0 does not vary continuously but is a "digital" variable, which assumes only discrete levels (e.g., +10 or −10 volts) corresponding to logical decisions. Timed or voltage-determined coefficient and computer-program changes are examples of such operations; their implementation requires *analog comparators* (elementary analog-to-digital converters, Sec. 4-9*a* and *b*) and *analog switching circuits* (elementary digital-to-analog converters, Secs. 5-6 to 5-9), together with suitable digital-logic circuits (see also Sec. 7-3).

SIMPLE LIMITER AND SELECTOR CIRCUITS

4-2. Junction-diode Characteristics and Diode Models.[1-3] An *ideal diode* would act as a perfect voltage-sensitive switch passing current in the forward direction only; its resistance would change without delay between zero and infinity as the applied voltage reverses. Figure 4-1*a* and *b* shows *voltage-current transfer characteristics* of an ideal diode and of a typical silicon-junction diode employed for analog-computer limiting and function generation. The ideal-diode characteristic is reasonably well approximated by the simplified current-voltage relation for silicon junctions,

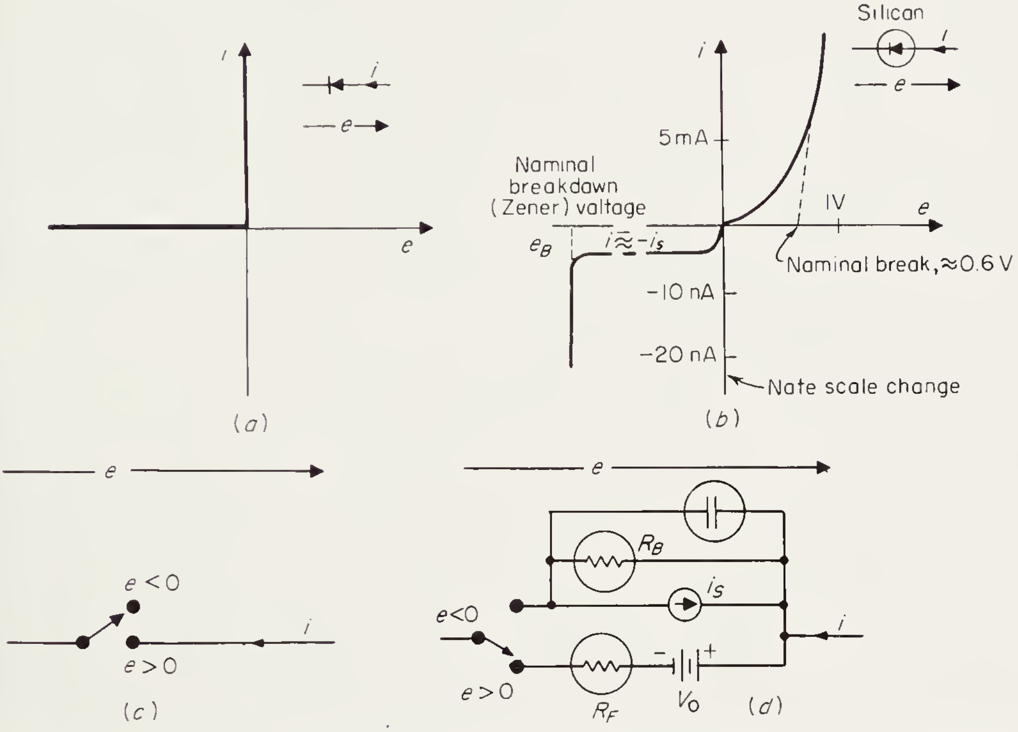


Fig. 4-1. Voltage-current transfer characteristics of an ideal diode (*a*), a silicon junction diode (*b*), and equivalent circuits (*c*, *d*).

$$i = i_s\left(\exp \eta \frac{q}{kT} e - 1\right) \qquad (e > e_B) \tag{4-1}$$

with $kT \approx 0.026$ volt at room temperature (25 deg C)

$$\eta = \begin{cases} 1 & (e \leq 0) \\ 2 & (e > 0) \end{cases},$$ to account for minority-carrier injection[1]

$i_s = 0.01$ to 10 nA (reverse saturation current inside the junction; doubles for every 7 to 10 deg C temperature increase)

Piecewise-linear analysis of the nonlinear diode characteristics (Fig. 4-1*c* and *d*) leads to various approximate equivalent circuits or *diode models* representing the behavior of real diodes in the ON and OFF conditions. In Fig. 4-1*d*, we have added a *leakage resistance* R_B (50 to 1,000M) to account for surface leakage and charge generation. The (nonlinear) *forward resistance* R_F includes the effect of bulk series resistance as well as that of the junction characteristic (1). Suitable linear or nonlinear capacitances are added to the diode model to approximate the dynamic (a-c) behavior of real junctions (Sec. 4-5).

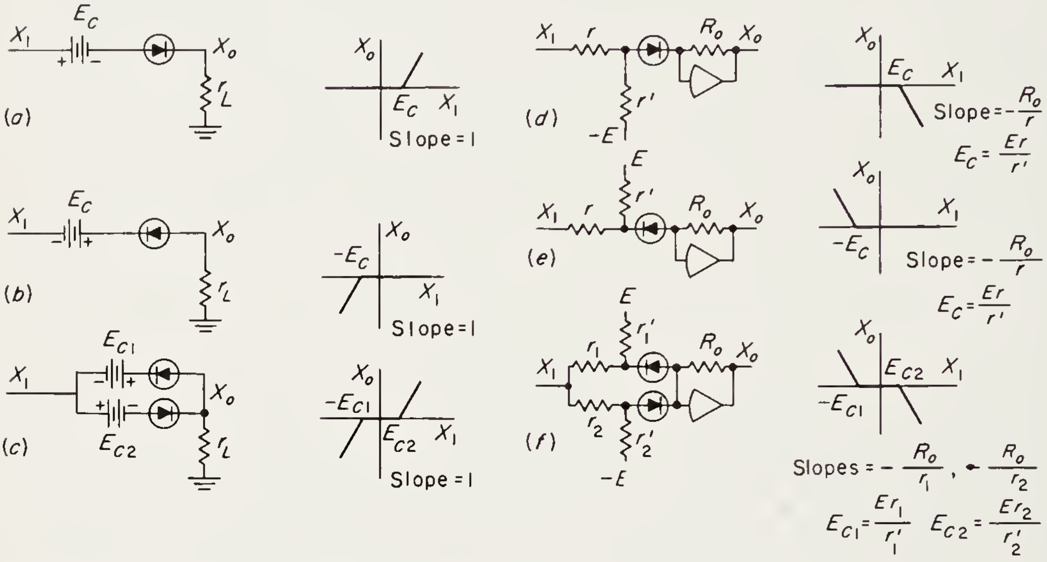


Fig. 4-2*a* to *f*. Basic series-diode limiters (*a*) to (*c*) and practical operational-amplifier circuits (*d*) to (*f*); transfer characteristics shown are for ideal diodes. E_C, E_{C1}, and $E_{C2} > -E_{C1}$ may be negative as well as positive, so that breakpoints can be positioned at will.

Since the true diode characteristic curves strongly near zero voltage, careful piecewise-linear approximation involves a *breakpoint voltage* different from zero (+0.5 to +0.7 volt for silicon junctions). The diode model of Fig. 4-1*d* includes this effect as a built-in reverse bias V_0 which is, in effect, added to any external bias voltage. Unfortunately, V_0 changes with temperature as determined by Eq. (1), so that we encounter *breakpoint drift*. Specifically, the built-in reverse bias of a silicon-junction diode *decreases* by 2 to 3 mV for each degree C of ambient-temperature increase. This effect requires either temperature control (component ovens) or compensation circuits (Fig. 4-12) for accurate computation.

For a sufficiently large reverse bias $e < e_B$, the junction-diode reverse current increases abruptly due to avalanche multiplication or (if e_B is less than 5 to 7 volts) to the zener effect.[1] *Breakdown* or *zener diodes* employ this second break in the diode characteristic for overload limiting (Fig. 4-6) and are also used in reference-voltage sources.

4-3. The Basic Diode Limiter Circuits.[6]

(a) The basic limiter circuits of Figs. 4-3 to 4-5 employ silicon-junction diodes to produce elementary nonlinear transfer characteristics. Simple limiter circuits can simulate important properties of dynamical systems (Secs. 8-18 to 8-20), and limiters can be combined to generate a large class of arbitrary functions (Sec. 4-15).

Limiter bias voltages can themselves be variables as well as constants. The series-shunt limiters of Fig. 4-5*a* and *b*, in particular, are useful as *amplitude selectors* producing the respective output voltages

$$X_0 = \min(X_1, E_c)$$
$$X_0 = \max(X_1, -E_c) \quad (4\text{-}2)$$

where E_c as well as X_1 can vary with time (see also Sec. 4-11).

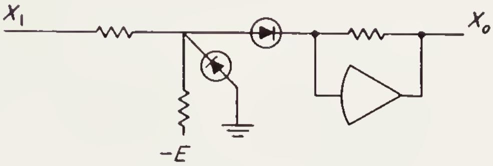


Fig. 4-3. Addition of a grounded "catching diode" reduces interaction between the sources of X_1 and $-E$ if their source impedances are higher than a few ohms, as in operational amplifiers operated at high frequencies. Catching-diode circuits combine shunt and series switching but should not be confused with the circuits of Fig. 4-5.

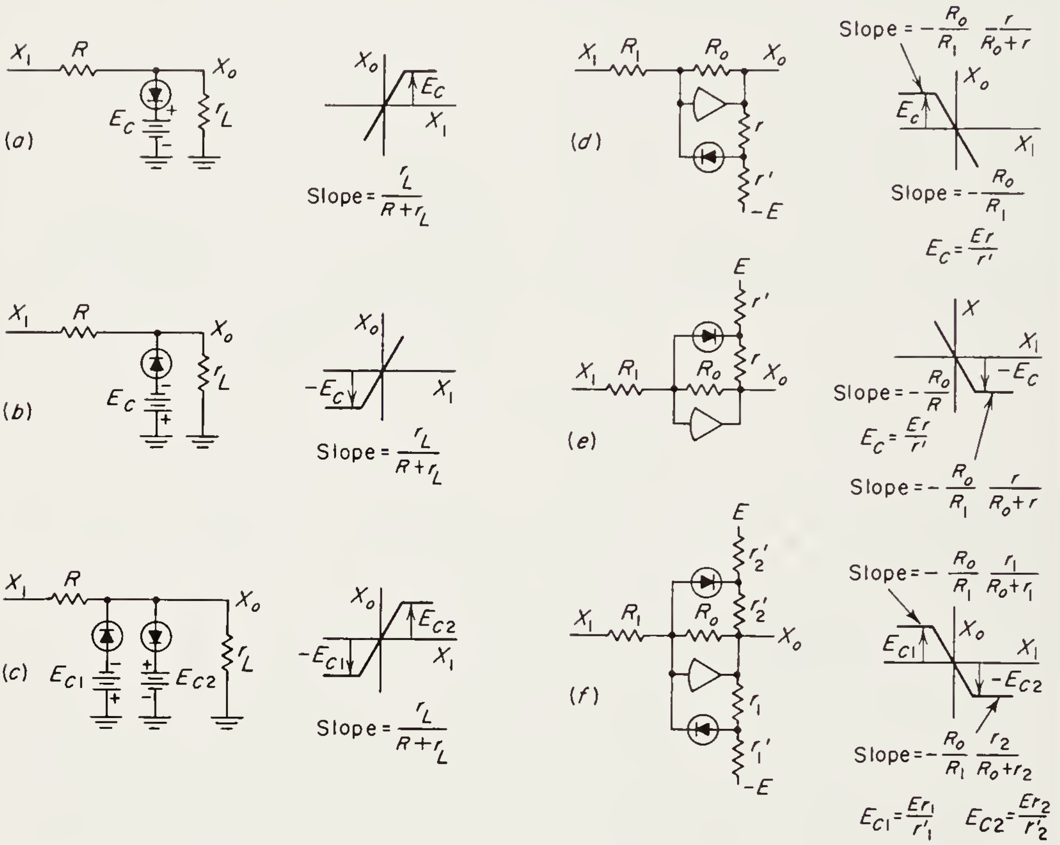
(b) Floating diode-bias supplies (Fig. 4-2*a*, *b*, *c*) are not practical in most computer systems. The more realistic circuits of Figs. 4-2*d* to 4-3 derive the required diode-bias voltages from grounded computer reference power supplies.

The accuracy of shunt and feedback limiters (Fig. 4-4) is usually compromised by bias-circuit impedances in series with the diode forward resistances, so that such circuits are more useful for easily adjustable overvoltage protection than for accurate computation. Silicon-junction-diode shunt limiters (Fig. 4-4*a*, *b*, *c*) can, however, limit more sharply if bias is supplied from a low-impedance source such as an operational amplifier rather than from the usual 20 to 100K potentiometer. More accurate diode limiter circuits are discussed in Secs. 4-6 and 4-10.

(c) *Zener-diode limiters* use the reverse-breakdown characteristics (Sec. 4-2) of specially designed diodes for limiting. Back-to-back zener diodes act like a pair of diodes with conveniently built-in fixed bias supplies (Fig. 4-6). Zener diodes can be obtained with a wide variety of breakdown voltages and can also be combined in series. Breakdown resistances are very low (typically between 10 and 100 ohms) and permit good shunt or feedback limiting if each diode is selected for a desired limiting voltage; the most useful application is, again, overvoltage protection (see also Secs. 4-8 and 7-6*b*).

4-4. Low-frequency Analysis of Diode Networks.

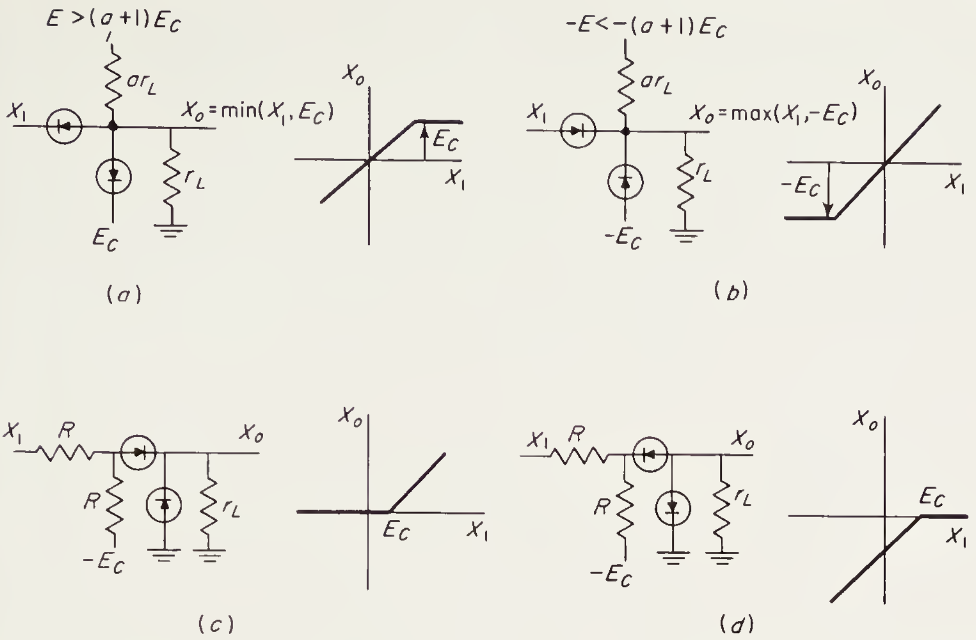
In many computer applications, diodes change the gain of a resistive network. One attempts to design these networks so that the finite forward and/or back resistances of physical diodes become negligible compared with other network resistances. With an occasional check on this condition, the simple ideal-diode model of Fig. 4-2*c* is very often satisfactory for analyses of "slow" analog-computer circuits employing silicon-junction diodes with $R_F \leq 50$ ohms, $R_B \geq 50$M, at frequencies below 2 to 20 Hz. The diode offset voltage, which could introduce errors of the order of 0.5 volt, can be disregarded when limiter breakpoints are set by empirical calibration; variations in the diode offset voltage with temperature (Sec. 4-1) will, of course, contribute to the computing error.



(a)
X_1
R
X_o
E_C
r_L
E_C
X_1
Slope $= \frac{r_L}{R+r_L}$
(b)
$-E_C$
(c)
E_{C1}
E_{C2}
$-E_{C1}$
E_{C2}
(d)
R_1
R_o
r
r'
$-E$
Slope $= -\frac{R_o}{R_1}\frac{r}{R_o+r}$
Slope $= -\frac{R_o}{R_1}$
$E_C = \frac{Er}{r'}$
(e)
E
$-E_C$
Slope $= -\frac{R_o}{R}$
(f)
r_2'
r_2
r_1
r_1'
Slope $= -\frac{R_o}{R_1}\frac{r_1}{R_o+r_1}$
$-E_{C2}$
Slope $= -\frac{R_o}{R_1}\frac{r_2}{R_o+r_2}$
$E_{C1} = \frac{Er_1}{r_1'}$
$E_{C2} = \frac{Er_2}{r_2'}$

Fig. 4-4. Shunt-diode limiters (*a*) to (*c*) and feedback-diode limiters (*d*) to (*f*); transfer characteristics shown are for ideal diodes. E_C, E_{C1}, and $E_{C2} > -E_{C1}$ may be negative as well as positive, so that breakpoints can be positioned at will. The limiting accuracy of these circuits is compromised by bias-circuit impedances (see also Sec. 4-4). Excellent shunt limiting at any desired level is possible if the low-impedance output of an operational amplifier is used as the bias source.

Figure 4-7 illustrates the static (d-c) piecewise-linear circuit analysis of the dual feedback limiter of Fig. 4-4*f* on the basis of our ideal-diode model. With the amplifier d-c gain between 10^4 and 10^8, it is fair to assume infinite d-c gain and zero output impedance. The circuit is



(a) (b)

(c) (d)

Fig. 4-5. Symmetrical series-shunt limiters shunt capacitive leakage, tend to cancel breakpoint drift, and are recommended for computing frequencies above 0.5 to 1 KHz. Low-capacitance germanium diodes are often used. E_C can be negative as well as positive; r_L can be the input resistor of an operational amplifier. Circuits (*a*) and (*b*) are supplied from low-impedance sources and have low output impedance in either state; they are useful as amplitude selectors, gates, and logic circuits. Additional diodes can be added to yield min $(X_1, X_2, \ldots)$ or max $(X_1, X_2, \ldots)$ (see also Sec. 4-11).

drawn and analyzed for each of its three possible states:

1. *D_1 and D_2* OFF (Fig. 4-7*b*):

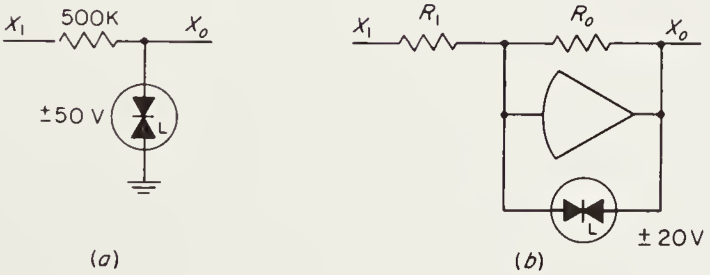
$$X_0 = -\frac{R_0}{R_1} X_1$$

2. D_1 ON, D_2 OFF (Fig. 4-7*c*):

$$X_0 = -\frac{R_0 r_1}{R_0 + r_1}\left(\frac{X_1}{R_1} - \frac{E}{r_1'}\right) \qquad \left(\approx -\frac{r_1}{R_1} X_1 + \frac{r_1}{r_1'} E \text{ for } r_1 \ll R_0\right)$$

3. D_1 OFF, D_2 ON (Fig. 4-7*d*):

$$X_0 = -\frac{R_0 r_2}{R_0 + r_2}\left(\frac{X_1}{R_1} + \frac{E}{r_2'}\right) \qquad \left(\approx -\frac{r_2}{R_1} X_1 - \frac{r_2}{r_2'} E \text{ for } r_2 \ll R_0\right)$$



(a) (b)

Fig. 4-6. Shunt and feedback limiters using back-to-back zener diodes.

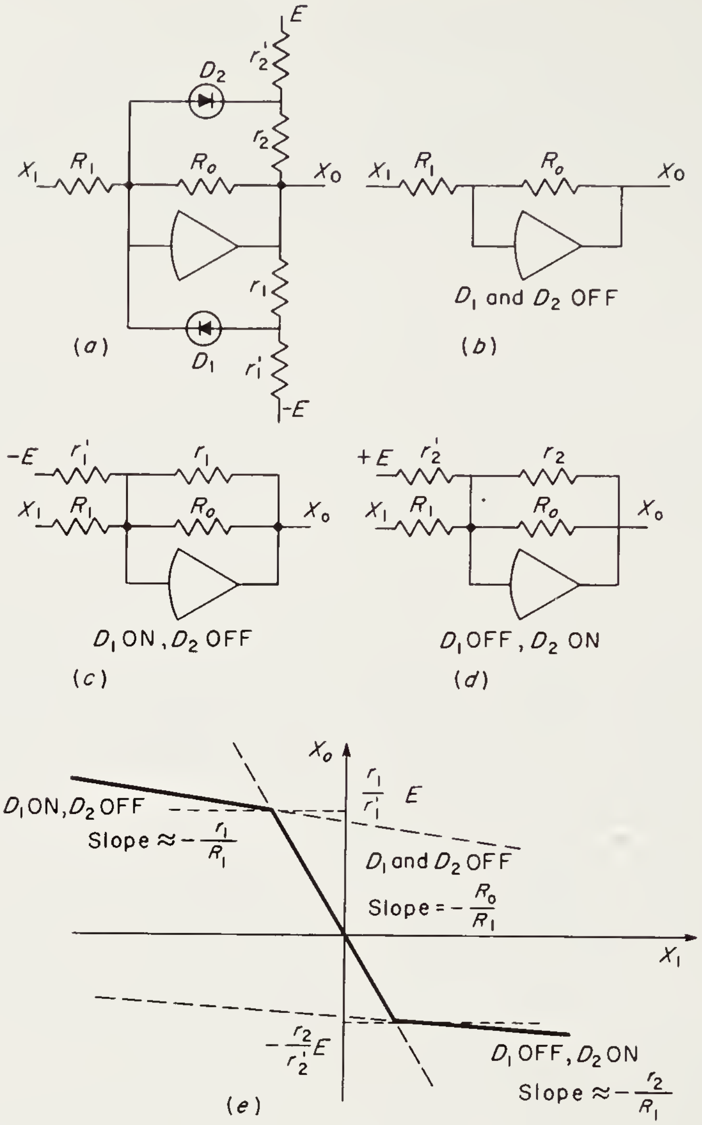
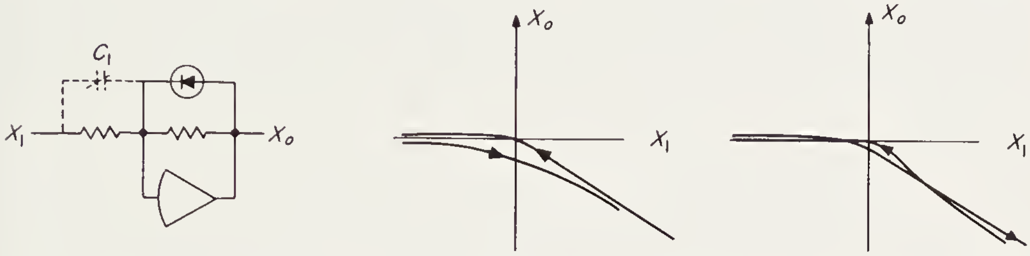


Fig. 4-7. Static piecewise-linear analysis of a typical limiter circuit (dual feedback limiter) on the basis of the ideal-diode model.

The fourth possible combination of diode states (D_1 and D_2 ON) cannot be realized, because the diodes are back-biased. In Fig. 4-7*e*, the three linear transfer characteristics are combined. To pick the correct sequence of line segments, one starts with the most negative input voltage and remembers that no ideal diode can carry reverse current.

The resulting transfer characteristic accurately represents the actual circuit performance, except in the immediate vicinity of the breakpoints. Note that the output still depends on X_1 even in the limiting conditions. For sharp limiting, r_1 and r_2 should be small, but small values of r_1, r_2, r_1', r_2' also increase the amplifier load current. The circuit limits more sharply at low limiting levels and is, in general, more suitable for overvoltage protection than for accurate limiting.

4-5. High-frequency Performance of Diode Circuits. In addition to the usual bandwidth limitations due to circuit capacitances and amplifier frequency response, the useful bandwidth of diode circuits is restricted by the effects of diode capacitances, which are highly nonlinear in junction diodes (Figs. 4-1*e* and 4-8*a*, *b*). The capacitance C_B of a junction diode reverse-biased by $-E$ is roughly proportional to $1/\sqrt{E}$. In addition, junction diodes exhibit nonlinear capacitance due to transit time and charge storage in the junction. The most noticeable effect is the *storage time* τ_s required to sweep minority carriers from the junction when the diode is turned OFF; this can be expressed in terms of a nonlinear capacitance C_F which increases with the forward current. While we can reduce the storage time by reducing the diode forward current, the charge-storage effect is usually masked by other capacitance effects if we employ fast-recovery diodes such as the 1N916A (OFF capacitance <4 pF; storage time <4 nsec from 10 mA at -6 volts).



(*a*) (*b*)

Fig. 4-8*a*, *b*. 1-KHz transfer characteristic of a feedback limiter (*a*), and effect of speedup capacitor C_1 (*b*).

Circuit and diode capacitances cause phase shift in each linear portion of a diode-circuit transfer characteristic. More seriously, circuit and diode capacitances must discharge reverse-bias voltages before each diode switches on; the resulting delays increase turn-on breakpoint voltages when the limiter-circuit input increases with time and decrease them for decreasing input. Dynamic errors, then, manifest themselves as a hysteresis-like spreading of diode-circuit transfer characteristics (Fig. 4-8*a*) which increases with frequency and with the input-voltage swing. The breakpoint shift (actually a time delay) is often roughly proportional to the rate at which the input voltage crosses the breakpoint. This effect becomes noticeable at frequencies well below those where junction-storage effects matter and is especially pronounced where the back resistances of open diodes can cause long discharge time constants. The spreading can be partially canceled by lead networks such as lead capacitors across operational-amplifier input resistors. The nonlinear nature of the circuit, however, usually permits compensation only over a portion of the working-frequency range and prevents accurate compensation near the breakpoints (Fig. 4-8*b*).

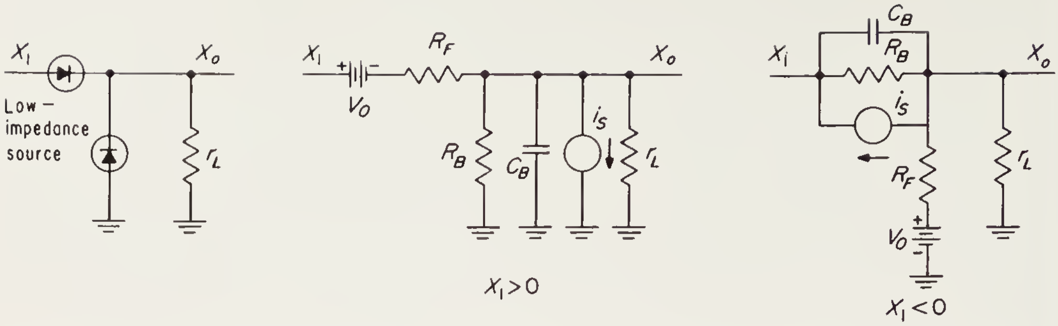


Fig. 4-8*c*. The equivalent circuit of a combined series-shunt limiter illustrates its beneficial effect on capacitive as well as resistive leakage. Turn-on and turn-off time constants are also low.

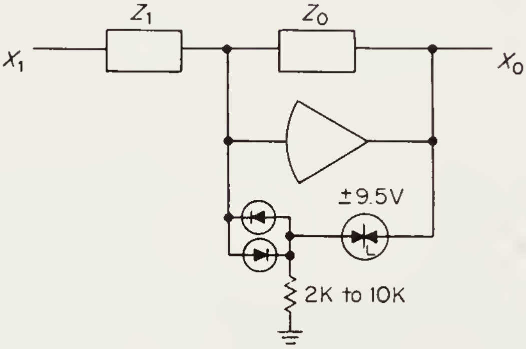


Fig. 4-8*d*. The 4- to 10-pF capacitances of simple feedback-limiter diodes can produce serious phase shift above 1 KHz. This *wideband feedback limiter* splits diode capacitances into T networks with easy ground paths for capacitive currents. This circuit is very important as a wideband overload limiter (Secs. 3-36 and 4-11) and in analog comparators (Sec. 4-9). This is probably the best of several similar circuits (Refs. 4 and 5).

Aside from the use of lead networks and fast-recovery diodes, one can reduce diode-circuit capacitance effects by keeping circuit impedance levels as low as amplifier power permits; note that this also requires the use of diodes with proportionally lower forward resistances. Substitution of diodes in series for single diodes or zener diodes can also decrease capacitance effects especially in feedback branches, if the increased forward resistance can be tolerated. With such precautions, the limiter circuits of Secs. 4-3 and 4-6 can be kept accurate within 0.5 percent of half-scale up to 200 Hz; each diode circuit will require separate investigation. At higher frequencies, the designer should attempt to "short-circuit" the worst capacitance effects by using *limiters and switching circuits which combine both shunt and series switching* instead of simple series, shunt, or feedback diodes (Figs. 4-5, 4-8*c*, and 5-1). Special circuits of this type permit accuracies within 0.3 to 1 percent of half-scale up to at least 100 KHz. The design of such circuits is still a challenging problem, and each case will require separate consideration (see also Secs. 4-15 and 4-17).

To study diode-capacitance effects on analog-computer accuracy, we can compute at a reduced time scale and vary artificially added capacitors across each diode.

4-6. Diode-bridge Limiters. The four-diode-bridge limiters shown in Fig. 4-9 produce accurate and conveniently adjustable dual-limiter characteristics by series-limiter action. Referring to Fig. 4-10, all four diodes conduct for low absolute values of the input voltage. When $|X_1| \geq Er/(r + 2r_L)$, the absolute value of the input-diode current $i = X_1(r + 2r_L)/2rr_L$ due to X_1 exceeds the input-diode current $I = E/2r$ due to the bridge-control voltages E, $-E$, and one of the input diodes (D_1 or D_3) cuts off. Note, however, that no limiting occurs at this point, since one pair of diodes (D_1, D_2 or D_3, D_4) is still conducting. Limiter action takes place when $|X_1| \geq Er/(r + r_L)$, so that one of the output diodes D_2 or D_4 is cut off as well. Figure 4-10*b* to *f* shows the resulting five possible states, and Fig. 4-10*g* and *h* illustrates the derivation of the input-diode currents i and I. The *half-bridge limiter circuits* of Fig. 4-11 will pass only input voltages of one sign.

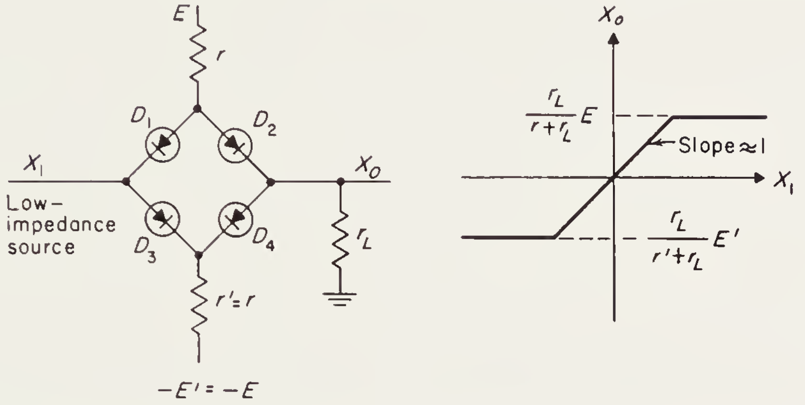
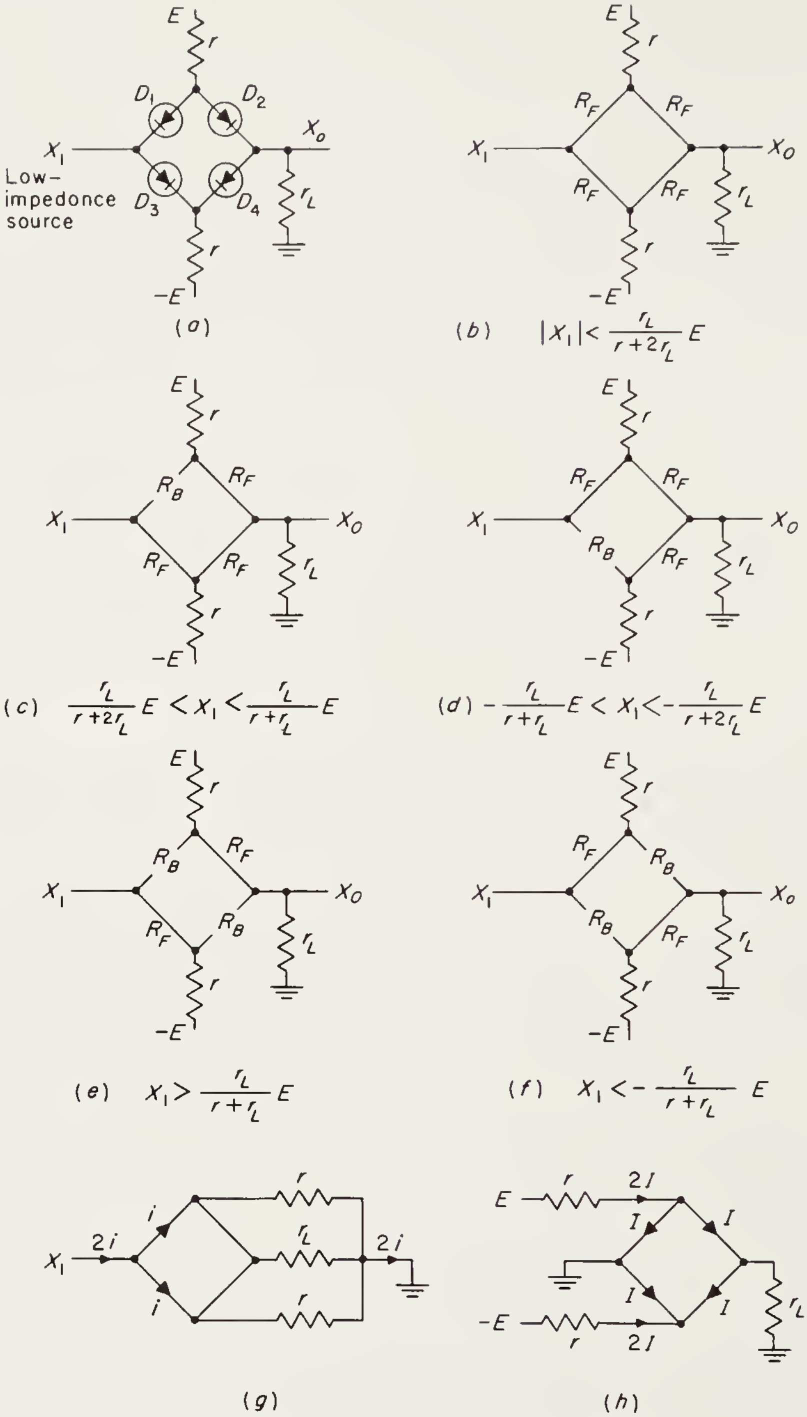


Fig. 4-9. Four-diode-bridge-limiter circuit. If desired, $r' \neq r$ and/or $-E' \neq -E$ yield asymmetrical limits, which must differ in sign. Variation of r_L (or a variable resistor shunting r_L) conveniently controls *both* limits.

Clipping levels are best set with the aid of a digital voltmeter or comparison potentiometer. Equivalent circuits like that of Fig. 4-10 show that, except in the immediate vicinity of the breakpoints, constant-temperature clipping levels are constant within 0.1 percent if $r_L/R_B \ll 1$ and any *changes* in R_F are within $10^{-3}r$; typically, $r = 100\text{K}$, $R_B = 100\text{M}$, and $R_F = 50$ to 500 ohms. The insertion loss associated with the diode-bridge limiter of Fig. 4-9 is also readily kept within 0.1 percent. Diode-bridge limiters can still limit within 0.5 volt at 1 KHz and are widely



(a)

(b) $|X_1| < \frac{r_L}{r+2r_L}E$

(c) $\frac{r_L}{r+2r_L}E < X_1 < \frac{r_L}{r+r_L}E$

(d) $-\frac{r_L}{r+r_L}E < X_1 < -\frac{r_L}{r+2r_L}E$

(e) $X_1 > \frac{r_L}{r+r_L}E$

(f) $X_1 < -\frac{r_L}{r+r_L}E$

(g)

(h)

Fig. 4-10. Four-diode-bridge limiter (*a*), its five possible states (*b* to *f*), and derivation of the input-diode currents i and I when all four diodes are conducting (*g*, *h*). D_1 cuts off when $i > I$, and D_3 cuts off when $i < I$. All formulas are derived for $R_F \ll r$, $R_F \ll r_L$.

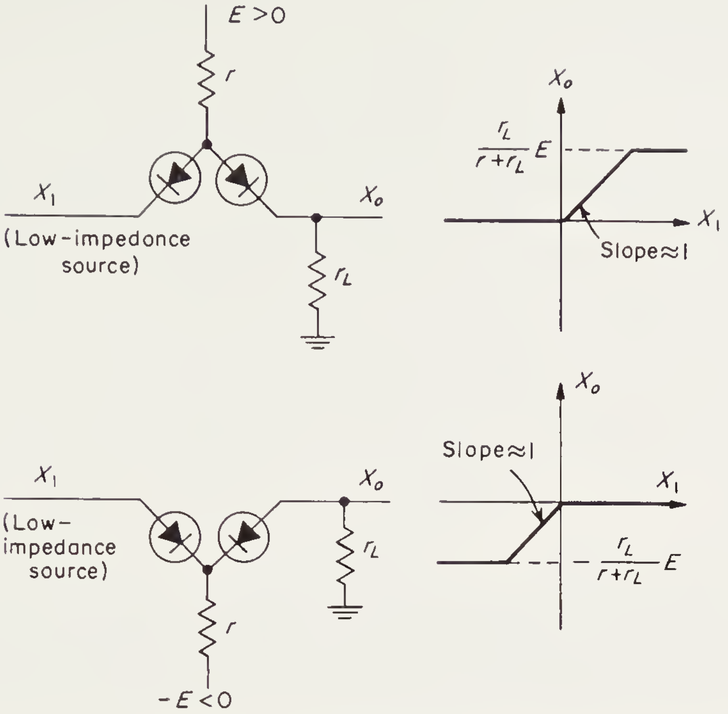


Fig. 4-11. Half-bridge limiter circuits and transfer characteristics. Each output voltage is limited between zero and a positive or negative voltage.

useful in computing and instrumentation applications. No multipurpose electronic analog computer should be without patchbay-mounted or plug-in bridge limiters. Refer to Sec. 5-2 for the use of diode-bridge circuits as switches.

4-7. Temperature Compensation of Breakpoint Drift (see also Sec. 4-15).[43–48] Figure 4-12 shows circuit designs which will compensate the 2 to 3 mV/deg C breakpoint drift (Sec. 4-2) of a diode limiter with the aid of an extra diode or diodes causing an opposing drift in the bias voltage. With carefully matched diodes (preferably monolithic sets), breakpoint-drift errors can usually be reduced by a factor of 5 to 10 over a 50 deg C temperature range. Such compensation circuits apply equally well to series limiters and feedback limiters (see also Figs. 6-4 and 6-28).

4-8. Transistor Limiters. Transistor limiters employ the cutoff characteristics of silicon-junction transistors (Sec. 5-3) for limiting. The limiting level is determined by the voltage between the transistor base and emitter (Fig. 4-13). Comparison of the *transistor feedback limiter* shown in Fig. 4-13*c* with the diode feedback limiter of Fig. 4-7 illustrates how transistor current amplification permits much sharper clipping especially when the limiting level differs from zero. For this

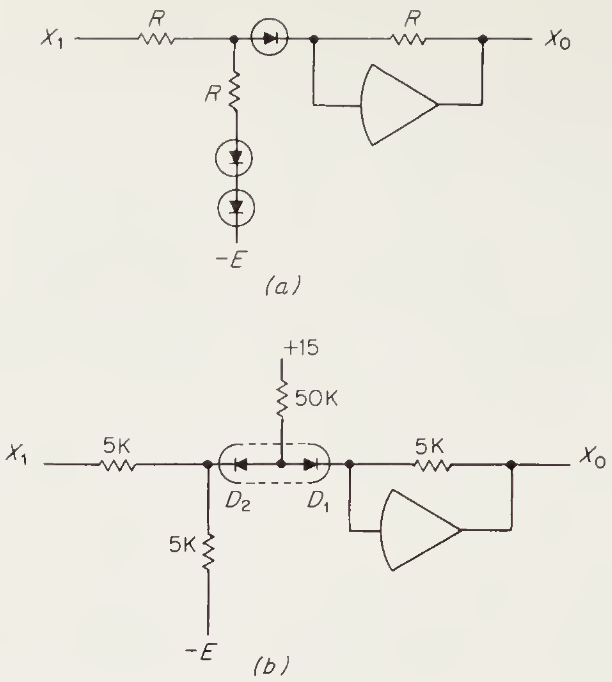


Fig. 4-12. Temperature compensation of diode-limiter breakpoint drift with matched diodes. Reverse diodes and bias for positive clipping. Note that the circuit of Fig. 4-12*b* is *not* used in the manner of a bridge limiter (Sec. 4-6); the compensating diode D_2 is always ON.

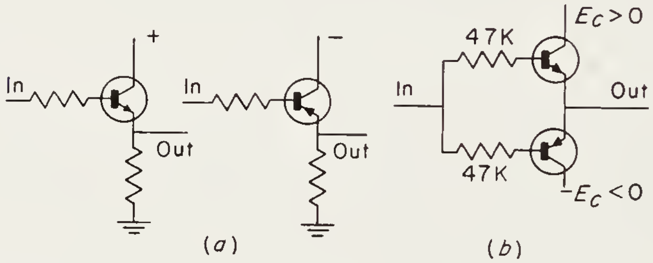


Fig. 4-13*a*, *b*. Simple transistor output limiters clipping at zero (*a*) and at $\pm E_C$ (*b*).

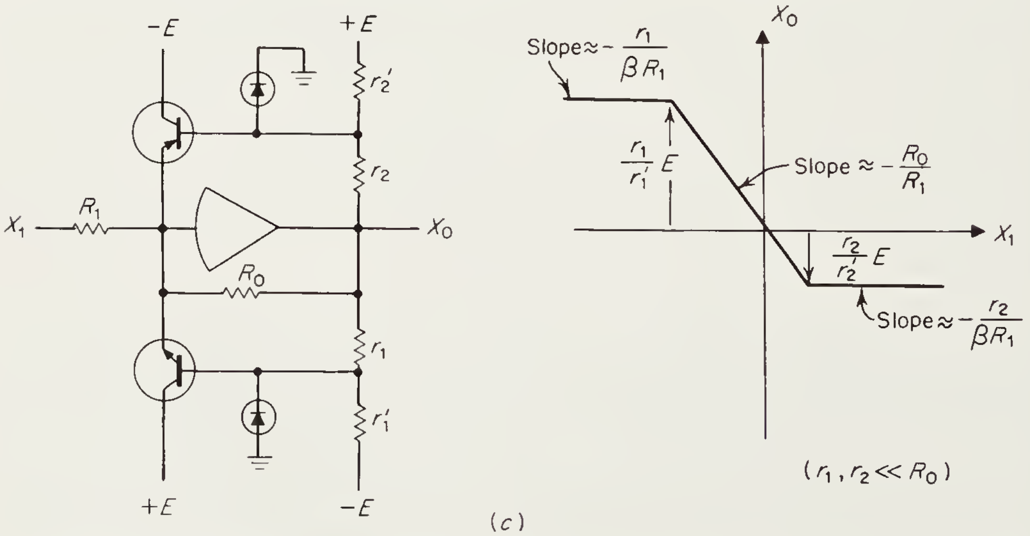


Fig. 4-13*c*. An adjustable transistor feedback limiter. Fast silicon transistors with high β (≥ 200) are used (e.g., 2N2222).

reason, limiter transistors should have current gains (beta) in excess of 200 (see also Fig. 6-28).

ANALOG COMPARATORS

4-9. Comparator Circuits. (a) Simple Operational-amplifier Comparators and Sign Multipliers. A high-gain operational amplifier without any feedback other than a feedback-limiter circuit has, except for brief transition periods, only two output states determined by the limiting levels (Fig. 4-14). The output voltage of such an *analog comparator* then changes decisively between two definite limit voltages whenever the sum of the input voltages changes sign: the comparator implements a *binary decision*, which can be interpreted as a basic 1-*bit analog-to-digital conversion*.

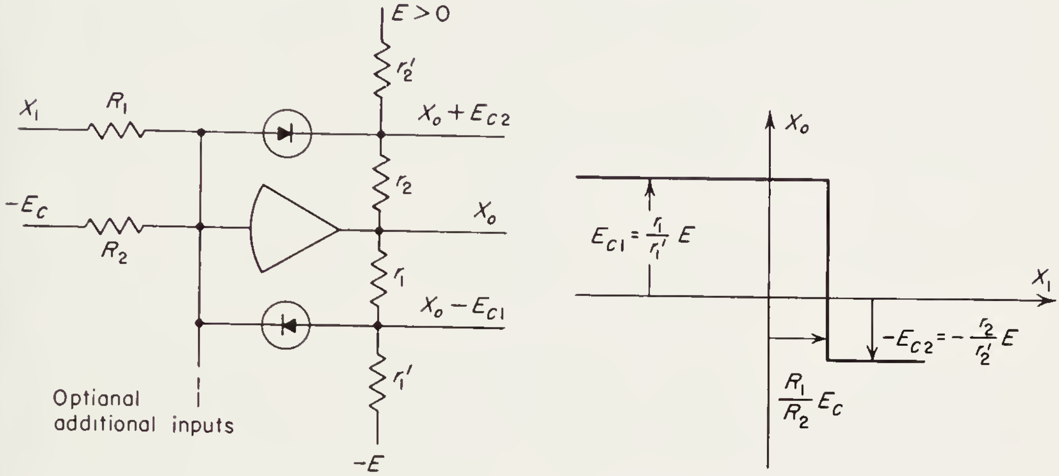


Fig. 4-14. A simple comparator and its transfer characteristic ($r_1, r_2 \ll R_1$). Transistor feedback may also be used. Such circuits can drive digital logic, electronic switches, relays, or lights. A diode bridge (Fig. 4-9) can follow the comparator to produce precisely defined output levels. The circuit shown supplies extra outputs $X_0 + E_{C2}$, $X_0 - E_{C1}$ which vary between 0 and $E_{C1} + E_{C2}$ and $-(E_{C1} + E_{C2})$, respectively, and can drive nonpolarized relays.

Figure 4-17 shows a useful single-input (zero-level) comparator circuit which provides precise output levels without diode feedback.

Analog comparators can drive indicator lights, relays, and electronic switches with or without intervening digital-logic circuits. Accurately determined comparator-output levels can also be used directly for analog computation. In particular, a comparator producing output levels X_2, $-X_2$ proportional to an analog input variable X_2 can produce the output

$$X_0 = -X_2 \operatorname{sign} X_1 \qquad (X_2 > 0) \tag{4-3}$$

(two-quadrant *sign multiplier*, Figs. 4-14 and 4-15; see also Sec. 8-9*b*).

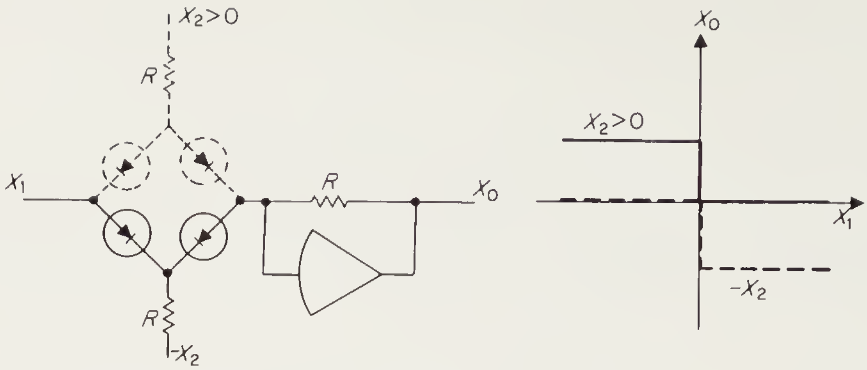


Fig. 4-15. This zero-level comparator has precisely defined output levels and is easy to patch on general-purpose analog computers. The reversed diodes and bias shown in dashed lines yield negative output for $X_1 > 0$; with the full bridge, one has a *sign multiplier* producing

$$X_0 = -X_2 \operatorname{sign} X_1 \qquad (X_2 > 0)$$

To remove the sign restriction on X_2, one can connect a second bridge with bias voltages $-X_2$, X_2 to the summing point (Refs. 31 and 92; see also Sec. 8-9*b*).

(b) Design of Improved Analog Comparators. For fast hybrid computation and analog-to-digital conversion, analog comparators must be designed for minimal *response times* to avoid timing errors (Fig. 3-2). The comparator response time will appear as the sum of an *output rise time* (which may be simply the rise time of a fast logic circuit connected to the comparator) and a *delay* required to charge circuit and feedback capacitances; the delay is roughly proportional to the time integral of the comparator input voltage, so that most practical comparators will switch more quickly for large and/or steep input-voltage transitions. Figure 4-16 shows a wideband comparator comprising a fast amplifier and a T-network diode-limiter circuit, which minimizes feedback through open diode capacitances (see also Sec. 4-5). In order to avoid jitter due to small noise inputs, regenerative feedback of a small portion of the comparator output voltage into the noninverting-amplifier

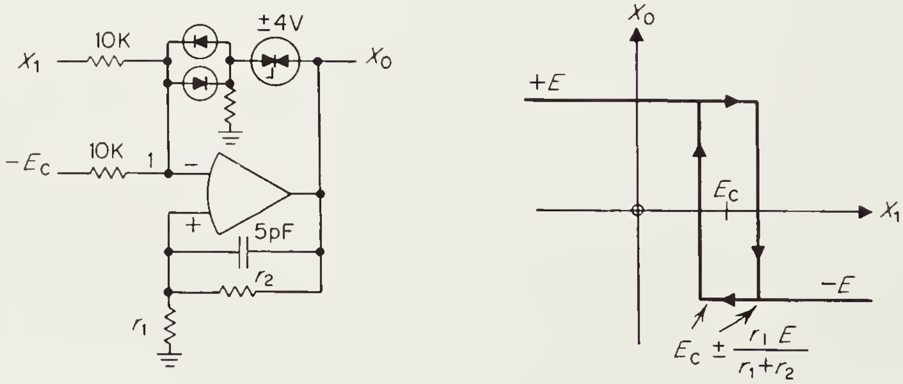


Fig. 4-16. A wideband comparator. Regenerative feedback produces hysteresis, which prevents output jitter due to small noise inputs.

input introduces *static hysteresis;* i.e., the regenerative feedback voltage biases the comparator so as to prevent output transitions due to small noise inputs (see also Sec. 4-12). Practical hysteresis levels are ±5 to ±20 mV, depending on the noise environment.

The Burr-Brown *adaptive comparator* incorporates a circuit which switches the hysteresis feedback off following input excursions greater than ±100 mV, so as to speed comparator response after such excursions; the jitter-reducing hysteresis is switched back on after the first small transition of the input voltage.

Feedback-limited comparators are gradually being replaced by special integrated-circuit comparators (such as Fairchild μA 710). These are very fast d-c amplifiers specially designed to recover quickly from overloads, so that no feedback limiting is required (Fig. 4-17).

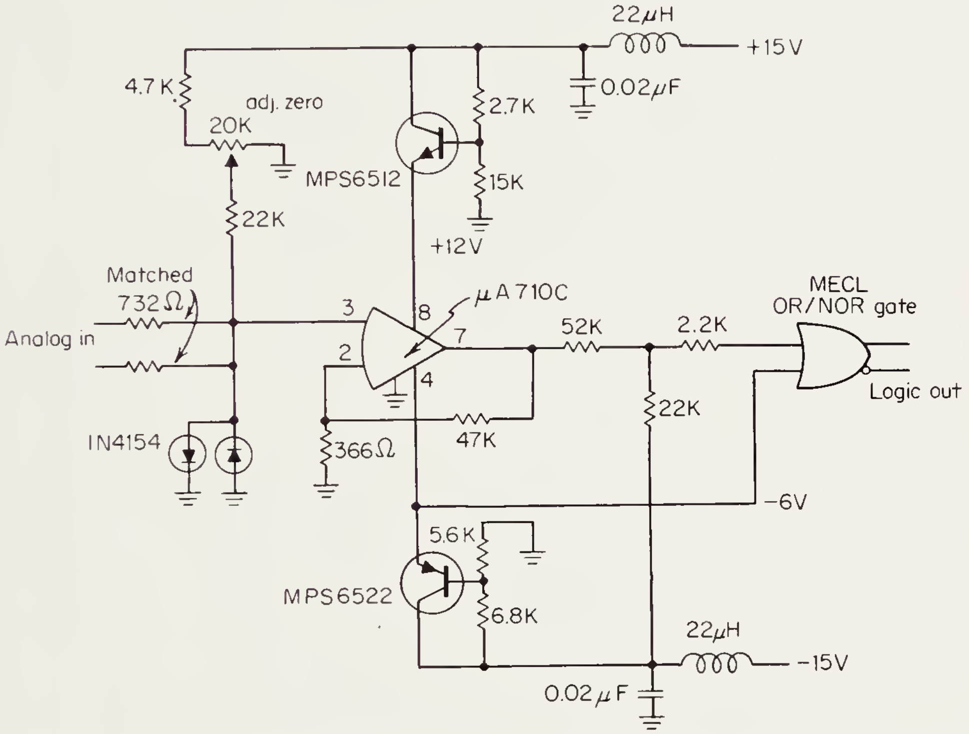


Fig. 4-17. A 100-nsec ±10-volt comparator with emitter-coupled-logic output. The +12-volt and −6-volt supply voltages for the μA 710C integrated circuit are obtained via emitter followers from the carefully decoupled ±15-volt analog-computer supply. Input diodes limit the μA 710C input to a safe ±0.7-volt, and very low input resistances prevent adverse effects from the offset-current drift of the inexpensive integrated circuit. Offset drift is thus within 100 μV/deg C at 25 deg C, with ±10 mV hysteresis. (*University of Arizona, Ref.* 94.)

Practical comparators have convenient logic-level outputs. They can also have an additional noninverting input, so that they may serve as *differential comparators* as well as with summing-network inputs. Fast comparators of this type may, however, have bias-current offsets (Sec. 3-32) as large as 0.1 mA/deg C and may not permit input voltages larger than ±5 volts. Figure 4-17 shows how a μA 710-type integrated circuit can be made into a computer-grade comparator with a low-impedance input circuit, input-protecting diodes, and hysteresis feedback.

Comparator level accuracy can be within a few millivolts, but hysteresis is practically always added in fast circuits. The comparator circuits of Figs. 4-16 and 4-17 permit response times between 50 and 200 nsec, depending on input signals and packaging. Still faster comparators can be built with tunnel-diode circuits, but they tend to be less accurate.

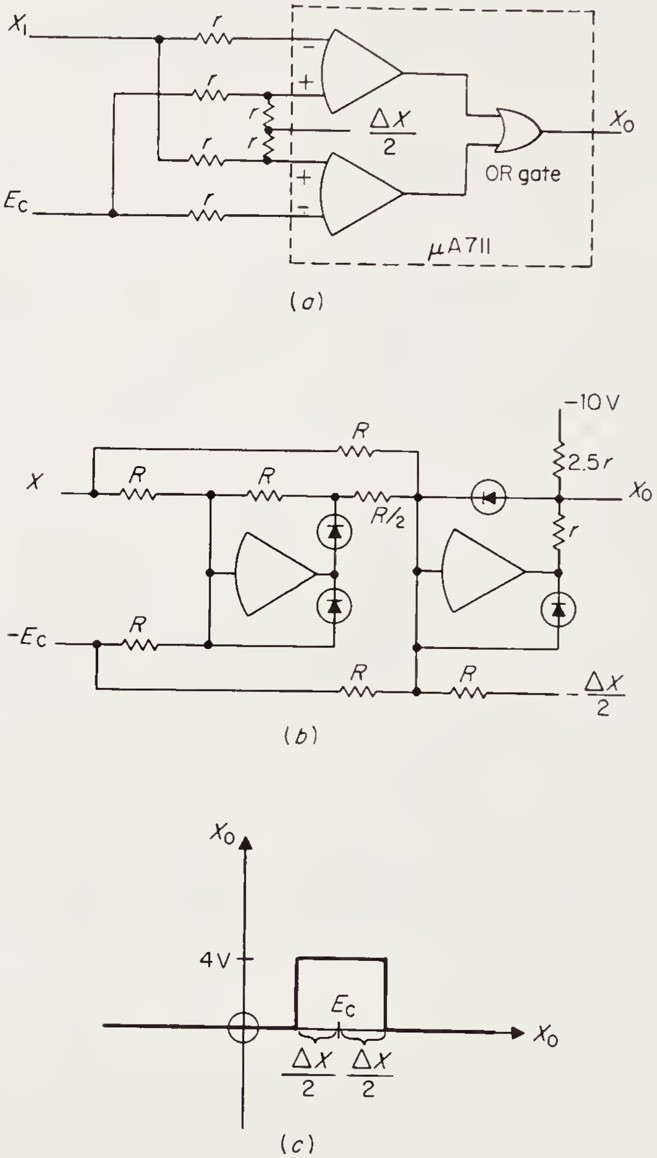


Fig. 4-18. A window comparator available in integrated-circuit form (*a*), an accurate window comparator based on the precision-limiter absolute-value circuit of Fig. 4-23 (*b*), and transfer characteristic (*c*). (*Refs*. 96 *and* 97.)

Fast output rise times required, e.g., for triggering fast flip-flops, are readily obtained from a fast logic gate or Schmitt trigger connected to the comparator output. Comparators may also be connected directly to set-reset flip-flops to provide "latching" or "memory." *Strobed comparators* have a gate input which enables the comparator when a timed strobe pulse is applied. Such pulsed operation can yield faster comparator response and may save a timed sample-hold circuit ahead of the comparator.

Figure 4-18 shows two *window comparators* which indicate whether the input sum is within an accurately designed class interval or "window." Such circuits are useful for go/no-go testing and for measurements of statistical relative frequency (amplitude-distribution analyzers).[96,97]

PRECISION LIMITERS AND SELECTORS

4-10. Precision Limiters and Some Related Circuits. **(a)** The basic diode-limiter circuits described in Sec. 4-3 yield static accuracies between 0.2 and 1 volt, depending on the particular application. By comparison, the *precision-limiter circuits* of Fig. 4-19 approximate ideal

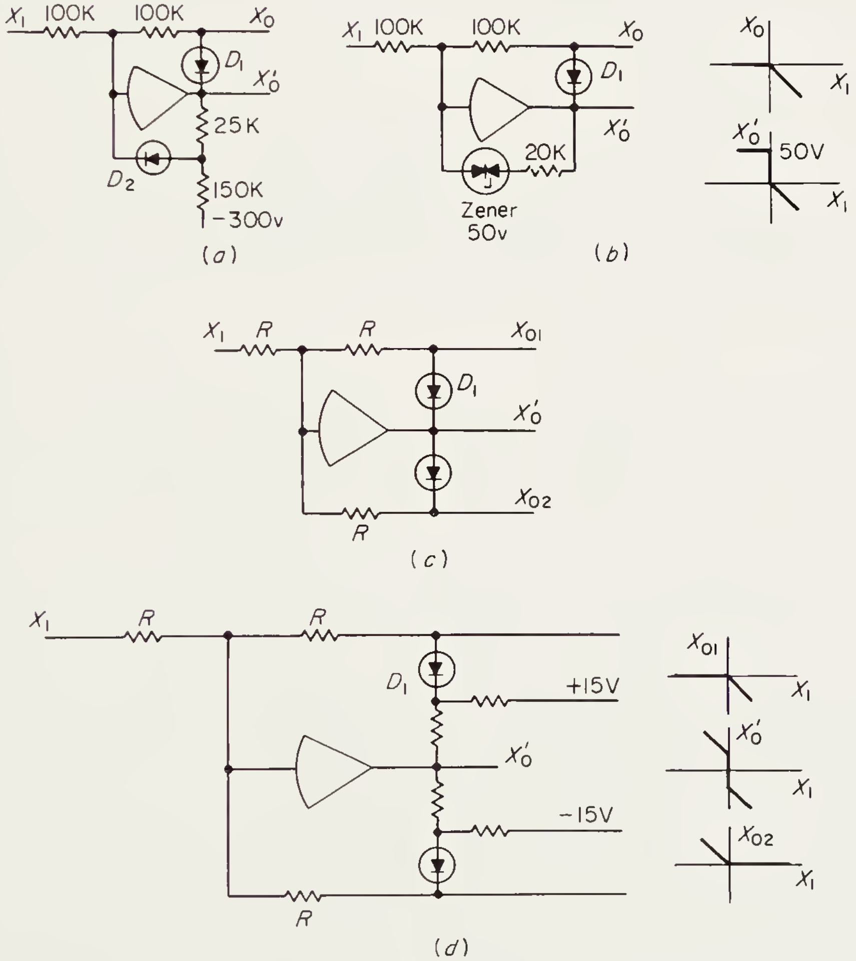


Fig. 4-19. Precision-limiter circuits. The circuits of Fig. 4-19*c* and *d* employ the "built-in" bias of a silicon-junction diode to produce a 0.4- to 0.7-volt comparator step; external bias, as in (*d*) however, can produce slightly better accuracy and also produces larger comparator steps to operate flip-flops, silicon controlled rectifiers, etc.

static limiter characteristics within 0.05 volt with the aid of high-gain low-drift d-c amplifiers.[16,17] When each diode D_1 in Fig. 4-19 conducts, it is inside a high-gain degenerative feedback loop, so that the nonlinear diode forward resistance and the "built-in bias voltage" (Fig. 4-1) are effectively divided by the loop gain. As D_1 ceases to conduct, the degenerative feedback decreases; the open-loop gain of the amplifier produces a decisive step in the voltage X_0' and cuts D_1 off with a very sharp break. Figure 4-20*a* compares a precision-limiter characteristic with ordinary series-limiter characteristics produced with silicon-junction and vacuum diodes. Such accurate nonlinear characteristics have a wide variety of applications in computation, instrumentation, and control.

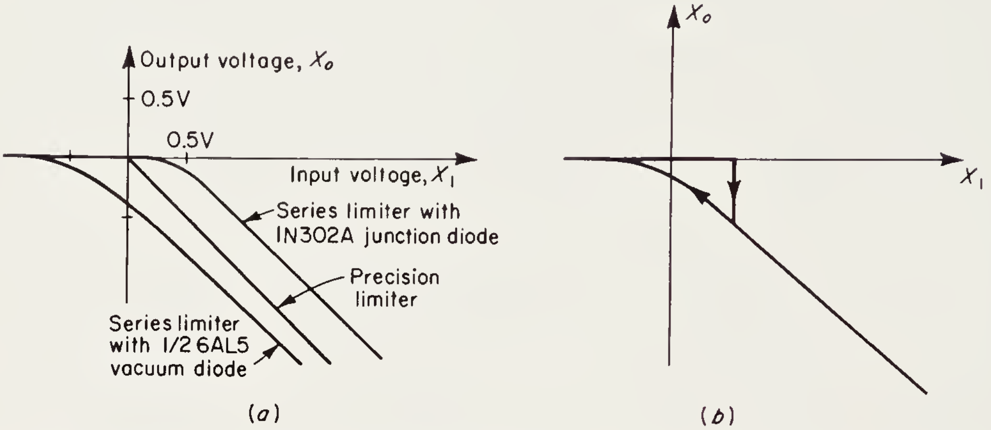


Fig. 4-20. Reproduction of an actual low-frequency (0.02-Hz sweep) servo-table plot comparing a precision-limiter characteristic and series-limiter characteristics obtained with the circuit of Fig. 4-2*d* using silicon-junction and vacuum diodes (*a*) and a precision-limiter characteristic taken at 5 KHz (*b*).

At higher frequencies, precision-limiter operation is modified by the combined effects of circuit and diode capacitances and amplifier rise time (Fig. 4-20*b*; see also Sec. 4-5). The phase-shift and comparator-step-delay errors illustrated in Fig. 4-20*b* increase with the input-voltage swing. High-frequency errors are minimized by fast amplifiers, low circuit impedances, and careful attention to low-capacitance layout. One further reduces the comparator-step delay by reducing the precision-limiter bias; the bias can be reduced to about 1 volt without affecting the static limiting accuracy. As with other diode circuits, speedup capacitors across the input resistor or resistors will improve the frequency response at least over a limited frequency range (Sec. 4-5). With low computing resistances (2K), suitable amplifiers, and fast diodes (1N916A), precision limiters can still yield 0.2 percent accuracy at 10 KHz.

(b) Figure 4-21*a* shows a precision limiter capable of limiting at a precisely adjustable positive or negative level other than zero (see also Sec. 4-11).[17] As in the precision-limiter circuits of Fig. 4-19, reversal of diodes and bias yields a reversed limiter characteristic (bottom limit instead of top limit). The *dual precision limiter* of Fig. 4-21*b*[17] again produces comparator steps to drive a diode bridge (Sec. 4-6) decisively

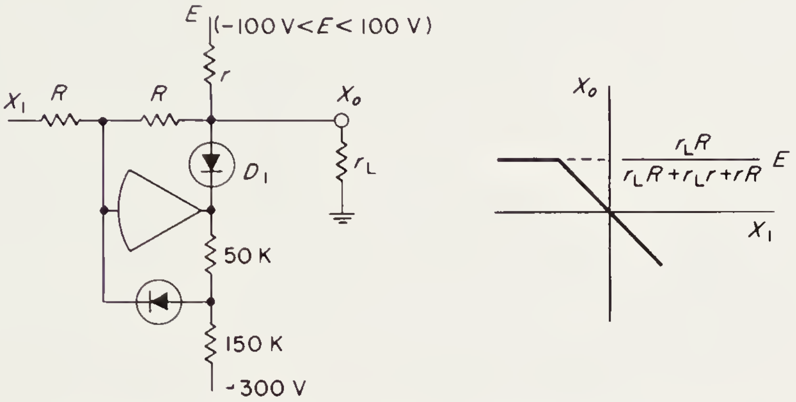


Fig. 4-21*a*. A precision-limiter circuit producing a nonzero limiting level. Reversal of diodes and bias yields a reversed (bottom-limited) characteristic.

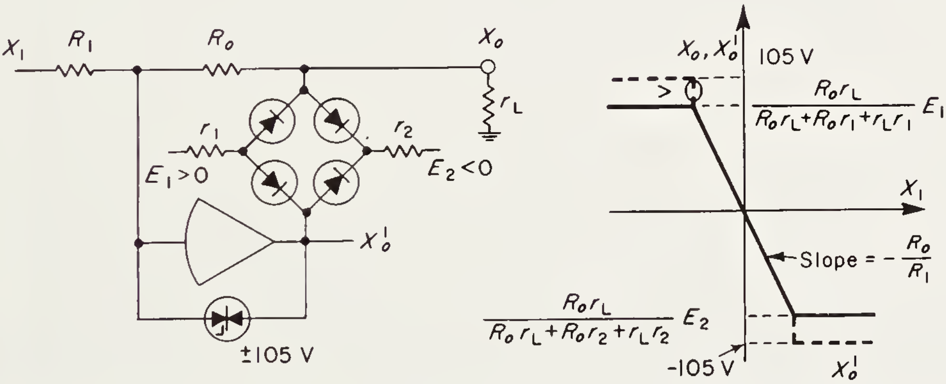

X₁
R₁
R₀
X₀
r_L
r₁
r₂
E₁>0
E₂<0
X′₀
±105 V
X₀, X′₀
105 V
>
$\frac{R_o r_L}{R_o r_L + R_o r_1 + r_L r_1} E_1$
X₁
Slope = $-\frac{R_o}{R_1}$
$\frac{R_o r_L}{R_o r_L + R_o r_2 + r_L r_2} E_2$
-105 V
X′₀

Fig. 4-21*b*. A dual precision limiter. The output X_0' again produces comparator steps to cut the diode bridge off decisively.

into its positive and negative limits. Because of the relatively large circuit capacitances involved, this circuit works best at frequencies below 50 Hz. As in Fig. 4-21*a*, the limit levels depend on the load resistance.

Figure 4-22*a* and *b* illustrates the use of diode-bridge feedback to produce useful *deadspace characteristics*. Figure 4-22*c* combines such a deadspace circuit with precision-limiter feedback to produce two

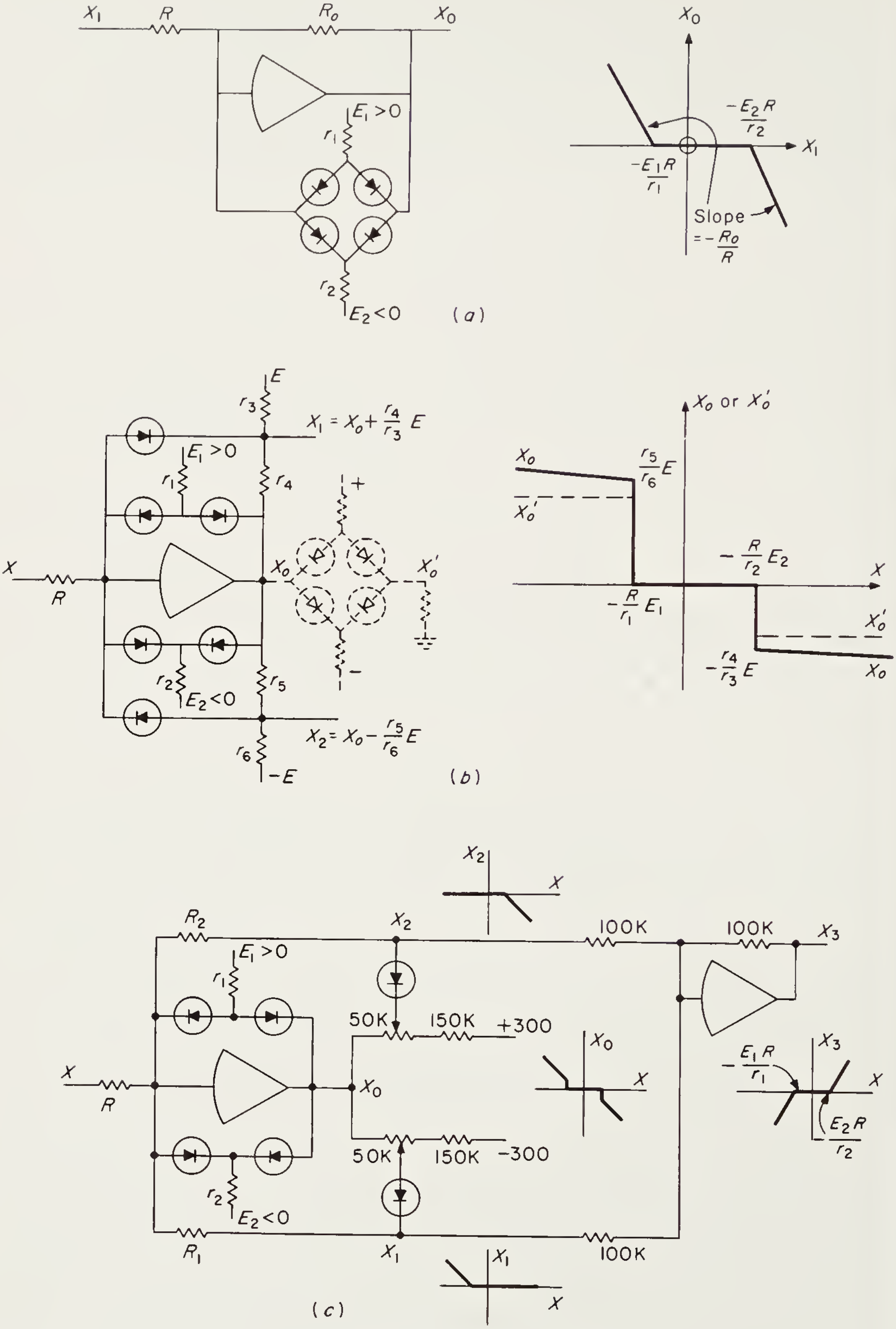


Fig. 4-22. Deadspace circuit using a diode bridge (*a*), deadspace comparator (*b*), and precision deadspace circuit (*c*).

precision-limiter outputs and comparator steps separated by a precisely adjustable deadspace. This *precision-deadspace circuit*[18] combines 0.05-volt static accuracy with useful operation up to at least 10 KHz. The circuits of Fig. 4-22*b* and *c* are not only useful for control-system simulation; they can perform automatic switching operations whenever the input voltage X_1 exceeds set limits, as required, for example, for

Go/no-go testing
Circuit protection
Automatic scale changes
Carry generation in hybrid computers
Counting control-transient cycles
Amplitude-distribution analysis
Resolver quadrant switching (Sec. 6-21)

Figure 4-23 shows the use of a precision limiter in a *precision absolute-value circuit.*

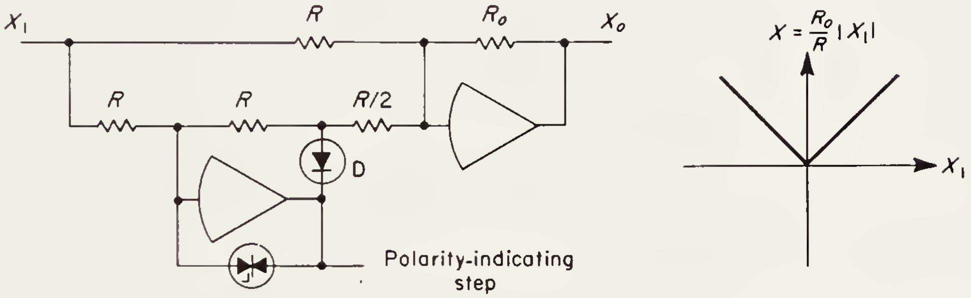


Fig. 4-23. Precision absolute-value circuit. Reversal of the limiter diode D yields $-(R_0/R)X_1$.

(c) Precision limiters (and related selector circuits, Sec. 4-11) can be regarded as basic "analog-to-hybrid-code converters." *Precision limiters produce not only very accurate analog limiter characteristics but also "digital" step outputs indicating the precise time when the input voltage crosses predetermined limits.* These step outputs are widely useful for actuating limit-indicating lights, relays, and/or digital-computer circuits.

While the precision limiters of Figs. 4-18 and 4-21 or 4-22, respectively, produce step outputs marking a single limiting level and two limiting levels, the "multicomparator" or quantizing circuit of Fig. 4-24[19] is a generalized precision limiter capable of marking 10 to 15 discrete voltage levels. Such a circuit can be useful for hybrid analog-digital computation, statistical sorting, or step-function generation. Figure 4-24*a* shows a high-gain d-c amplifier with nonlinear feedback through a diode function generator having the transfer characteristic (Y vs. X_0) defined by Fig. 4-24*b*. The resulting feedback is alternately degenerative and zero (or very slightly regenerative) as the input voltage X_1 changes. It follows that the forward transfer characteristic (output X_0 vs. input X_1, Fig. 4-24*c*) exhibits sharp comparator steps at accurately defined input-voltage values; the regenerative action will tend to make the breakpoint values of X_1 independent of diode characteristic and temperature.

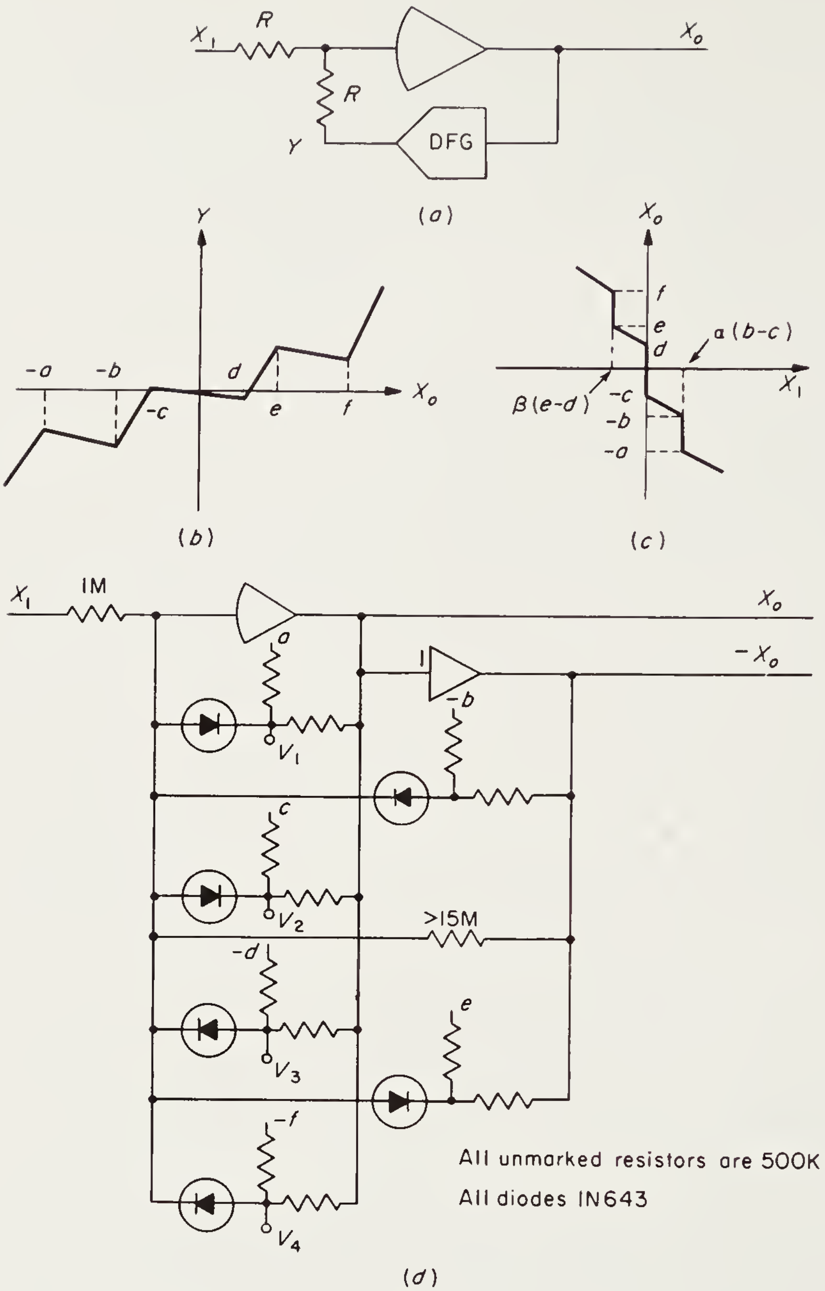


Fig. 4-24. The multicomparator (*a*) employs alternating degenerative and very slightly regenerative feedback (*b*) to produce a transfer characteristic with multiple comparator steps (*c*). Figure 4-24*d* shows a practical circuit which, in addition to its multicomparator outputs X_0, $-X_0$, produces individual comparator steps V_1, V_2, V_3, V_4.

Figure 4-24*d* illustrates a practical circuit. *Note that the output terminals marked* V_1, V_2, V_3, V_4 *yield positive or negative comparator steps starting at zero*, which is particularly useful for operation of associated digital circuitry. Addition of extra diode channels yields up to 15 comparator steps ($16 = 2^4$ states); $2n$ diodes yield $n + 1$ states, and only two amplifiers are used. Unity slopes and equal breakpoint intervals are shown for simplicity, although other values may be preferable for various applications. Too much regeneration will cause comparator steps with hysteresis.

4-11. Accurate Selectors, Extreme-value-holding Circuits, and Peak Detectors. The precision-limiter circuit of Fig. 4-25[17] functions as a *precision selector* which produces the larger of two constant or variable voltages X, Y. Reversal of diodes and bias would yield min (X,Y) instead of max (X,Y). Unlike most precision-limiter circuits, this two-amplifier selector has a low source impedance in *both* of its two switching states.

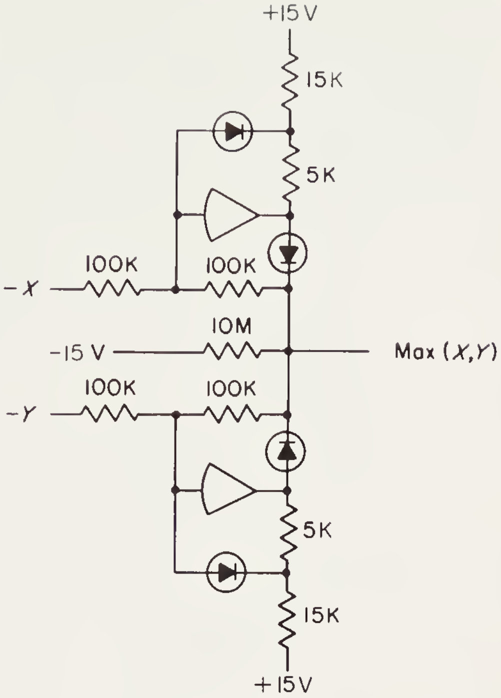


Fig. 4-25. A precision-selector circuit producing the larger of two voltages X, Y. Reversal of diodes and bias would yield min (X,Y) instead of max (X,Y). The outputs of up to at least five precision limiters with inputs $-X_1$, $-X_2$, . . . can be similarly connected together to produce max $(X_1,X_2, \ldots)$ or min $(X_1,X_2, \ldots)$.

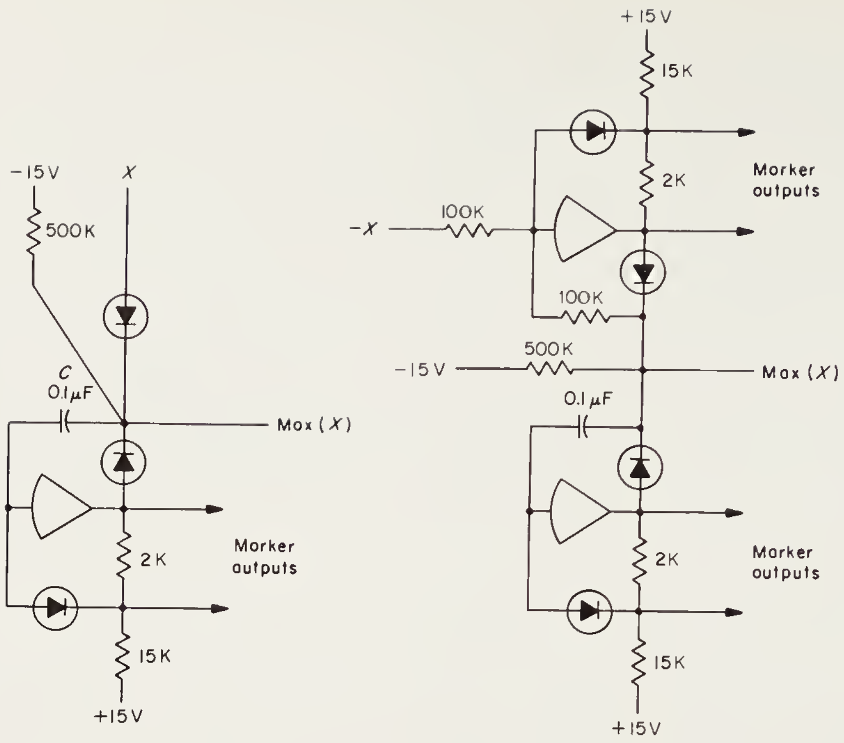
Each of the *maximum-holding circuits* of Fig. 4-26[20] combines a precision selector with an integrator. These circuits mark successive relative maxima with precisely timed digital steps and hold the largest past value of the input X. These circuits were originally developed for spacecraft applications (rocket-thrust measurements) and are also useful in many simulation problems. If the storage capacitors are shunted with suitably large resistances, the same circuits become accurate *peak detectors* for a-c waveforms.

BISTABLE OPERATIONAL-AMPLIFIER CIRCUITS AND SPECIAL SIGNAL GENERATORS

4-12. Multivibrator Circuits. Figure 4-27*a* shows a simple analog bistable circuit.[17,21] The comparator biases itself through a phase-inverting amplifier, so that an input voltage equal to or larger than $+E_C$ is needed to change the output state. The input voltage then loses control until it reaches the negative value $-E_C$.

The addition of an integrator to the basic multivibrator circuit of Fig. 4-27*a* yields a *free-running* (*astable*) *multivibrator* (Fig. 4-27*b*). Figure 4-27*c* shows a *monostable multivibrator circuit* capable of delivering precisely timed pulses after triggering.

The astable multivibrator of Fig. 4-27*b* is widely useful as an accurate and versatile signal generator producing various periodic waveforms



(*a*) (*b*)

Fig. 4-26*a*, *b*. These maximum-holding circuits accurately store the largest past value of a voltage $X = X(\tau)$ and produce sharp comparator steps marking successive relative maxima.

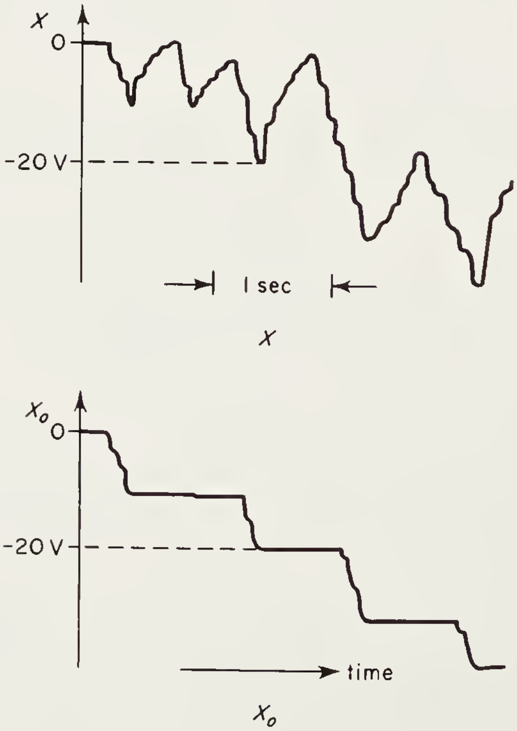


Fig. 4-26*c*. Record produced by a *minimum-holding* circuit obtained from Fig. 4-26*a* or *b* by reversal of diodes and bias. A 0.1-μF holding capacitor was used.

with analog-computer accuracy. With the addition of a simple −90 to +90 deg sine-generating diode network, one obtains an accurately timed sine wave as well as triangular-wave and push-pull square-wave or pulse outputs (Fig. 4-28*a*). The triangular waveform is invaluable for measuring transfer characteristics of function generators, transistors,

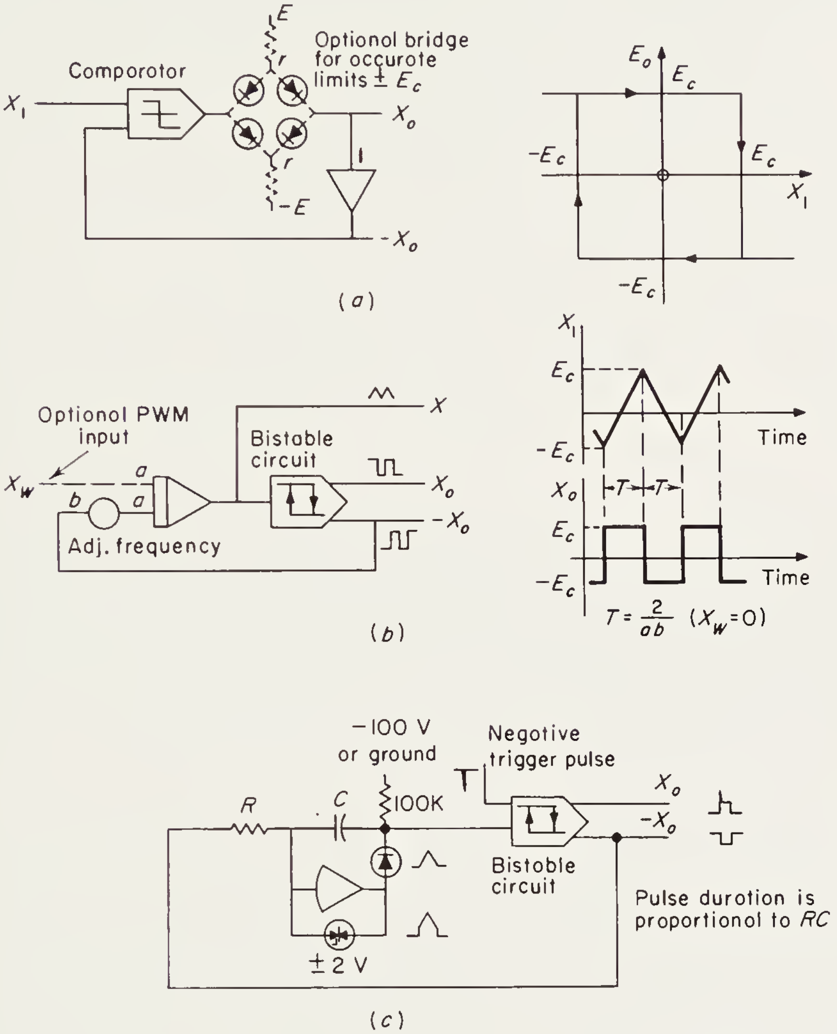


Fig. 4-27. Analog bistable element with push-pull output (*a*), astable (free-running) multivibrator (*b*), and monostable multivibrator (*c*). If only one accurate analog output $-X_0$ is needed, we can interchange the positions of bridge limiter and inverter in Fig. 4-27*a* and substitute an inexpensive one-stage inverter for the precision feedback inverter without any loss in accuracy. We can also vary the amount of hysteresis by changing the regenerative loop gain.

etc., at different frequencies, and for testing multipliers (Sec. 6-15). The pulse outputs can operate electronic switches, relays, lights, or digital computing elements, and pulse differentiation yields useful phase-reference markers. Figure 4-28*a* to *c* also shows how easily various accurately modulated signals are produced by multivibrator-type signal

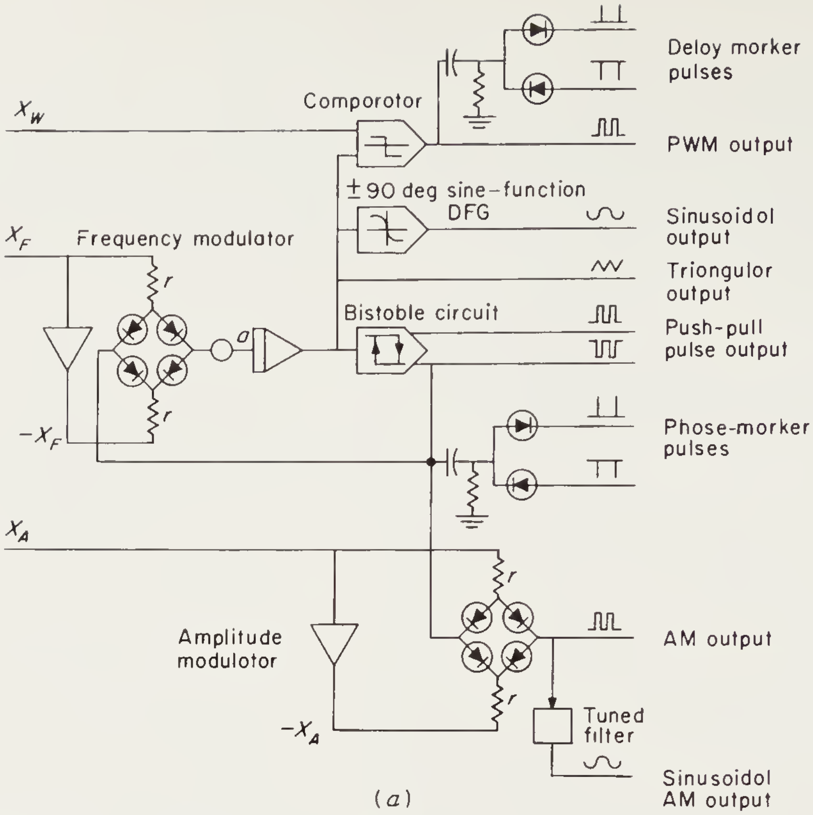


Fig. 4-28a. Generation of different modulated waveforms with the simple free-running multivibrator of Fig. 4-27*b*. All outputs permit electrical frequency modulation completely independent of their amplitude or pulse-width modulation. Depending on the components used, the frequency range between 0.0001 and 10 Hz can be covered with timing and amplitude accuracies within 0.1 percent of half-scale; accuracies within 1 percent of half-scale can be obtained up to 50 KHz. Clearly, only the portion of the circuit specifically needed for a given application need be built or patched.

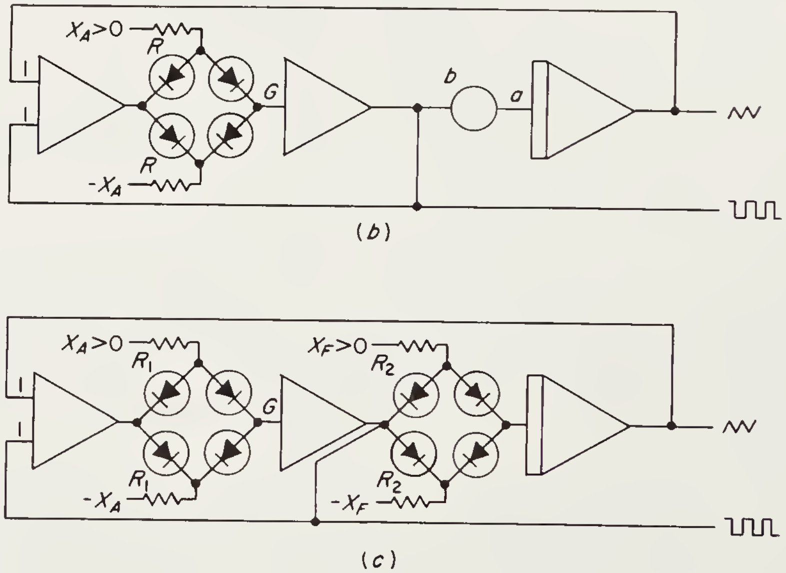


Fig. 4-28*b*, *c*. Two accurate astable multivibrators using the fast comparator circuit of Fig. 4-15. Figure 4-28*b* permits amplitude modulation of both triangle-wave and pulse output at constant frequency. Figure 4-28*c* provides for frequency modulation at constant amplitude, or for amplitude modulation at constant triangle slope (see also Refs. 30 and 31).

generators. Frequencies all the way between 0.005 Hz and 500 KHz can be obtained with suitable amplifiers.

Low-frequency signal generators of this type are easily patched on general-purpose analog computers; the resulting signals will be more accurate than those produced by most commercially available test instruments. Permanently connected analog-computer bistable circuits used in special-purpose equipment need not always employ expensive chopper-stabilized d-c amplifiers. Depending on the particular application, only one or two amplifiers may require drift stabilization, or a-c coupling may be possible.

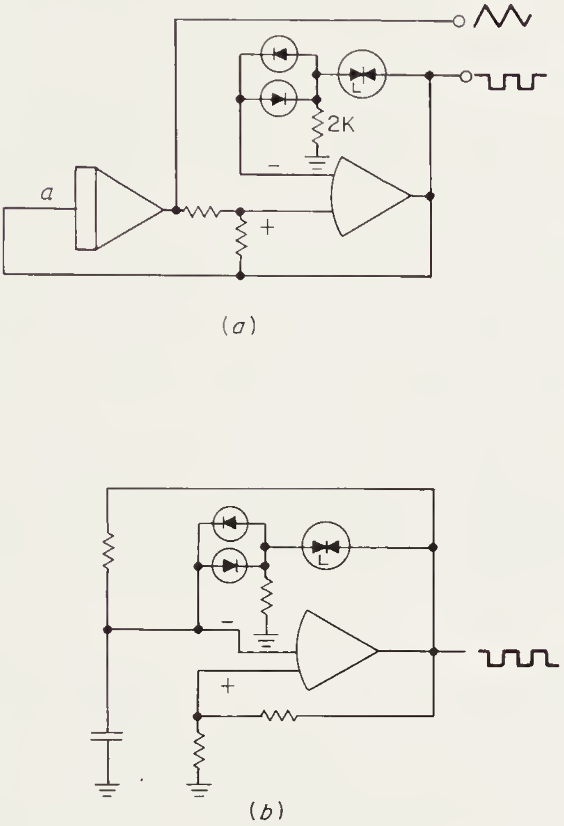


Fig. 4-29. Simplified operational-amplifier-multivibrator signal generators.

Figure 4-29 shows two simpler astable multivibrator circuits.

4-13. Accurate Sine-wave Oscillators. Multivibrator/function-generator-type oscillators are probably the most widely useful general-purpose signal generators. Direct generation of sinusoidal waveforms is, however, preferred if segmented sinusoidal output cannot be tolerated, if the square-wave and triangular multivibrator waveforms are not needed, and/or where both quadrature outputs $\pm a \sin \omega t$ and $\pm a \cos \omega t$ are required, as in servomechanism testers and for plotting circles or ellipses.

An *induction resolver rotated by a simple rate servomechanism* at a manually adjustable angular velocity ω yields 60 Hz to 10 KHz output modulated by $\pm a \sin \omega t$ and $\pm a \cos \omega t$. A second induction resolver manually rotated through a desired phase angle ϑ can be used to produce "variable-phase output" modulated by $a \cos (\omega t + \vartheta)$ for null-type phase measurements. The resulting modulated test signals serve as inputs to 60- or 400-Hz control-system components, or a phase-sensitive demodulator is used to yield low-frequency sinusoidal output.

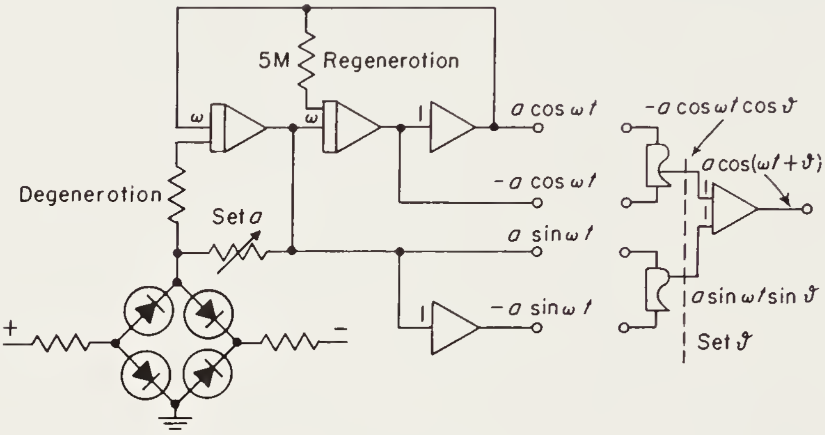


Fig. 4-30. An ordinary differential-analyzer sine-generating loop yields $\pm a \sin \omega t$, $\pm a \cos \omega t$, and with a manually adjusted dual sine-cosine potentiometer, $a \cos (\omega t + \theta)$. A diode-bridge limiter controls the oscillation amplitude; better accuracy is possible with an AGC circuit like that in Fig. 4-31*b*.

All-electronic sine-cosine generators produce voltages $\pm a \sin \omega t$, $\pm a \cos \omega t$, and $a \cos (\omega t + \vartheta)$ directly. They are smaller and lighter, require less maintenance, and have a vastly wider frequency range than servo-type signal generators. Since *LC* oscillators do not easily yield low distortion and heterodyne signal generators tend to have poor frequency stability, sine-wave oscillators operating in the frequency range between 0.001 Hz and 50 KHz are *RC phase-shift oscillators.*

Figures 4-30 and 4-31 show different types of *RC* phase-shift oscillators employing inexpensive computer-type d-c amplifiers to permit accurate operation at frequencies all the way between 0.01 Hz and 2 MHz. If output below, say, 10 Hz is not required, d-c coupling and chopper stabilization are, of course, unnecessary, but utility-type plug-in d-c amplifiers are still convenient and yield high gain to reduce distortion and output impedance. Each phase-shift oscillator comprises a regenerative linear feedback loop with phase shift slightly in excess of 180 deg at the oscillator frequency. The resulting oscillation must be limited to a stable amplitude with a *nonlinear amplitude-control circuit.* The simplest circuit of this type is a simple feedback limiter or bridge

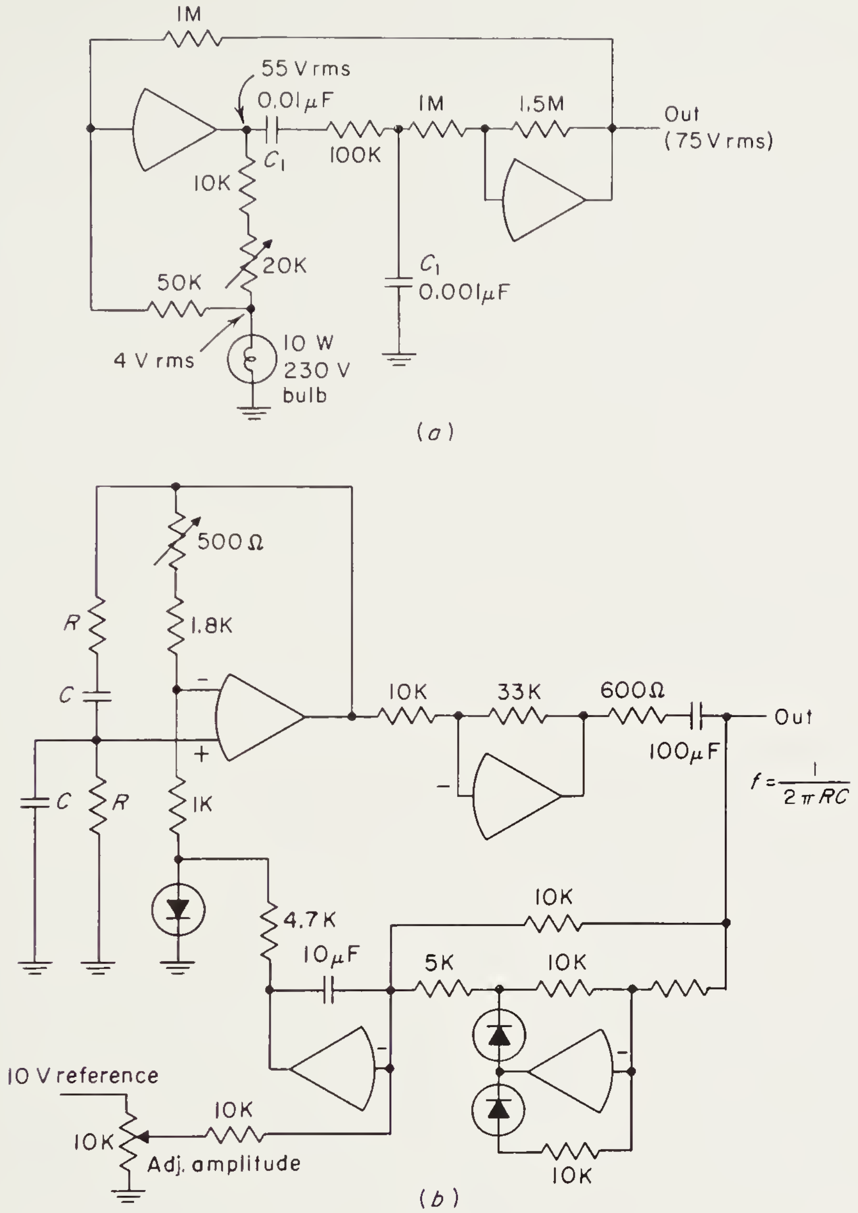


Fig. 4-31. Wien bridge oscillators employing utility-type ±100-volt d-c amplifiers (*a*), and ±10-volt d-c amplifiers (*b*). A practical frequency-range switching scheme might employ the same capacitor as C_1 and as $C_2 = 0.1\ C_1$ in different frequency ranges. Arrows indicate where quadrature voltages 90 deg out of phase with the main oscillator output could be taken off with the aid of additional amplifiers. Very accurate amplitude-control circuits require a-c coupling, chopper stabilization, or low-drift differential-input transistor amplifiers. The circuit of Fig. 4-31*b* yields amplitude stability within ±0.02 db, frequency stability within ±1 percent, and less than 0.5 percent distortion between 100 Hz and 6 KHz. (*Burr-Brown Research Corporation, Tucson, Ariz.*) The amplitude-regulating diode in Fig. 4-31*b* could be replaced by a *p*-channel junction FET with the control voltage applied to the gate.

limiter in the sine-generating loop (Fig. 4-30); the resulting distortion can be kept as low as 1 percent of the amplitude if the sine-generating loop is adjusted to be only very slightly regenerative. Reference 26 describes a similar *three-phase oscillator* with three sinusoidal outputs.

Improved sine-wave accuracy requires automatic gain control throughout the cycle instead of simple clipping. Medium-accuracy oscillators operating above 20 Hz have long employed the nonlinear resistance of a tungsten-filament lamp or other temperature-sensitive resistor to control the oscillation amplitude (Fig. 4-31). Such simple amplitude-control circuits involve a compromise between distortion and amplitude stability, especially if the oscillator is to operate over a wide frequency range.[24,25] At the lower frequencies, temperature-sensitive resistors tend to follow the instantaneous oscillator output instead of its rms value, and nonlinear distortion results; at higher frequencies, increasingly linear operation may permit amplitude jitter due to hum and line or load transients. Temperature-sensitive resistors should also be operated at temperatures high enough to render the effects of ambient-temperature variations negligible.

More recent types of low-voltage transistor phase-shift oscillators replace the tungsten-lamp amplitude control by the nonlinear resistance of a silicon-junction diode or junction FET biased by an AGC voltage.[32,33] The diode is operated in the knee of its characteristic. Figure 4-31*b* shows an amplified-AGC or integral-control scheme. The AGC voltage is obtained through precision full-wave rectification of the sinusoidal oscillator output (Sec. 4-10); peak detection could be used instead. Such circuits permit amplitude control within 0.1 percent with 0.25 percent distortion down to at least 20 Hz.[25]

Without question, the most accurate low-frequency sine-cosine oscillator is a two-integrator sine-generating loop with continuous minimization of the amplitude error $|\varepsilon| = |\sin^2 \omega t + \cos^2 \omega t - 1|$ through feedback into one or both integrators, as in Fig. 6-34*b*. Such circuits can yield amplitude accuracy and distortion within 0.05 percent but are relatively expensive.

ELECTRONIC FUNCTION GENERATORS

4-14. Introduction (see also Sec. 4-1). The simplest analytic functions of a voltage (machine variable) X are polynomials like $aX + b$, aX^2, $X^3 + bX^2 + cX$, etc.; such functions are readily produced by combinations of summers, multipliers, and coefficient-setting potentiometers. Simple differential-analyzer setups can produce analytic functions of the computer time τ (e.g., $ae^{b\tau}$, $a \sin b\tau$; Fig. 1-4*d*) and, if we introduce generalized integration through multiplication by $dX/d\tau$, also similar functions of a voltage $X = \int_0^\tau (dX/d\tau)\, d\tau$ (see also Sec. 1-13).

More general types of functions must be set up on *general-purpose function generators;* most electronic analog computers employ *diode function generators* (see also Sec. 4-1). Function-generator accuracy and frequency response will depend on the nature of the function generated. We can often reduce the effects of function-generator errors substantially by using the function generator to correct a linear or analytic approximation function obtained with accurate summers, potentiometers, and multipliers. In addition, every effort should be made to utilize the full dynamic range of function-generator input and output variables by efficient scaling (Sec. 1-12).

Figure 4-32 shows a circuit producing the *inverse function* $Y = G(X)$ of a computer-generated function $F(Y)$. The circuit continuously enforces the relation

$$X - F(Y) = 0 \qquad (4\text{-}4)$$

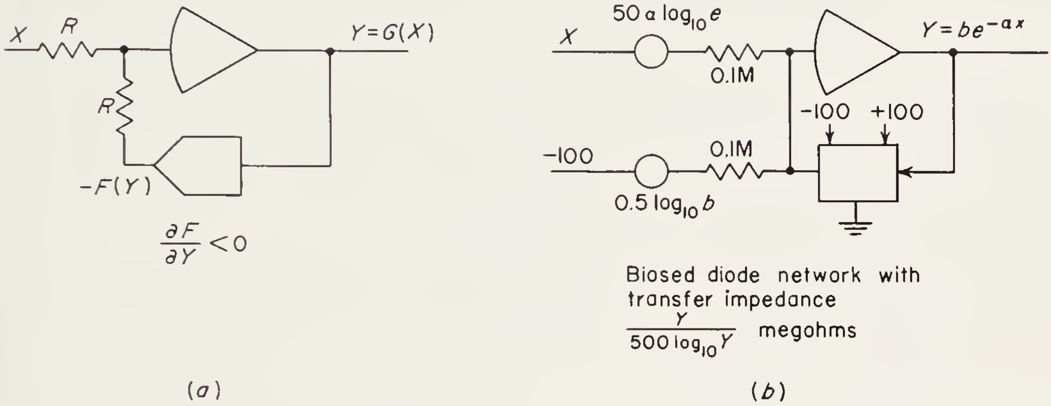

X
R
Y=G(X)
R
-F(Y)
$\frac{\partial F}{\partial Y} < 0$
(a)
50 a log$_{10}$ e
X
0.1M
Y=be^{-ax}
-100 +100
-100
0.1M
0.5 log$_{10}$ b
Biased diode network with transfer impedance $\frac{Y}{500 \log_{10} Y}$ megohms
(b)

Fig. 4-32. Generation of inverse functions by implicit computation (see also Sec. 8-9). Figure 4-32*b* illustrates generation of $X_0 = be^{-\alpha X_1}$ ($b \geq 1$) with a commercially available logarithm-generating diode network.

where the condition $\partial F/\partial Y < 0$ is necessary to ensure stability of the implicit-computation loop; high-frequency stability requires separate investigation. The initial value of Y must equal the correct initial value of $G(X)$ (see also Sec. 8-9).

4-15. Diode Function Generators. (a) Special-purpose Function Generators. Smoothing Oscillators. The broken-line transfer characteristics of resistive networks containing biased diodes can approximate a large class of functions. Nonmonotonic functions must be obtained by addition of a nondecreasing function and a nonincreasing function. Diode function generators permanently set up for aX^2, $a \sin bX$, and $a \log_e X + b$ are very useful in many applications of general-purpose analog computers (Secs. 4-16 and 6-1). Fixed-function generators are useful in many control and instrumentation applications.

Figure 4-33 illustrates the use of a high-frequency (5 to 500 KHz) *smoothing oscillator* for rounding diode-function-generator characteristics near the breakpoints.[39] The triangular or sinusoidal smoothing-oscillator voltage is added to the function-generator input, which is thus

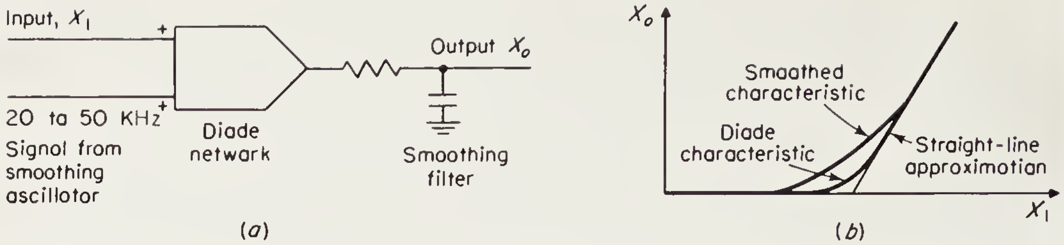


Fig. 4-33. Diode function generator with smoothing oscillator to smooth diode-network transfer characteristics (*a*), and smoothed limiter-channel characteristic (*b*).[39]

swept across the breakpoints. An output averaging filter yields the smoothed characteristic and, unfortunately, also introduces low-frequency phase shift (see also Sec. 6-12).

In fixed-function generators, junction-diode breakpoint drift due to temperature changes can be partially compensated by thermistors, by extra junction diodes in the bias networks (Figs. 4-12, 6-4*b*, and 6-28), or by series-shunt diode pairs (Fig. 4-8*c*). Diode bias compensation schemes are discussed in detail in Refs. 43 to 48. The advent of low-cost integrated-circuit operational amplifiers makes it also possible to employ multiple "active" precision limiters (Sec. 4-10) in function generators.

(b) General-purpose Function Generators. General-purpose diode function generators sum the output currents of between 5 and 24 simple

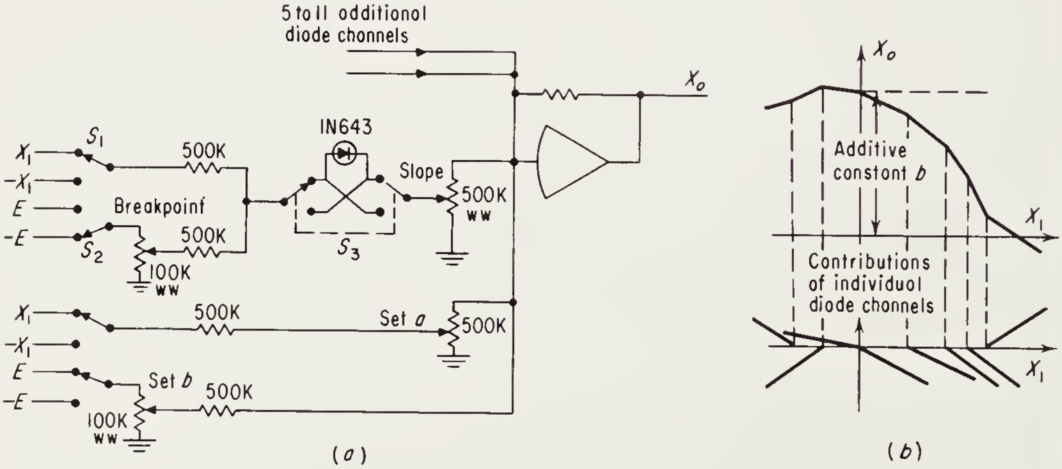


Fig. 4-34. This Reeves type (series-limiter) diode function generator requires only a single patchbay-connected amplifier but has both X_1 and $-X_1$ inputs. The switches S_1, S_2, S_3 in each channel can be combined into a four-position "quadrant switch" yielding only the four types of segments shown in Fig. 4-34*b*, but this restriction is not necessary. 10-turn potentiometers set individual slopes and breakpoints with a minimum of interaction; note that the slope-adjustment scheme used loads the input much less than a set of slope-adjusting potentiometers ahead of the 500K summing resistors would. The two lower channels add a linear function $aX_1 + b$. E is usually the computer reference voltage; if E is replaced by a second variable voltage Y, the device produces various functions of two variables (Sec. 4-19*b*).

limiter channels with adjustable breakpoints and slopes to generate straight-line-segment approximations of different functions (Figs. 4-34 and 4-35). Natural breakpoint curvatures afford a measure of slope interpolation between line segments, and smoothing oscillators can be used if desired.

A single limiter-output-summing amplifier can yield only monotonic functions. To generate nonmonotonic functions, we can supply some of the limiter channels with $-X_1$ instead of X_1 to obtain limiter slopes of either sign (Fig. 4-34), or we can use two amplifiers to subtract two monotonic functions (Fig. 4-35). The latter method is often used with permanently committed amplifiers for wideband operation. Two diode-function-generator networks can be combined by patching if a given function requires many breakpoints.

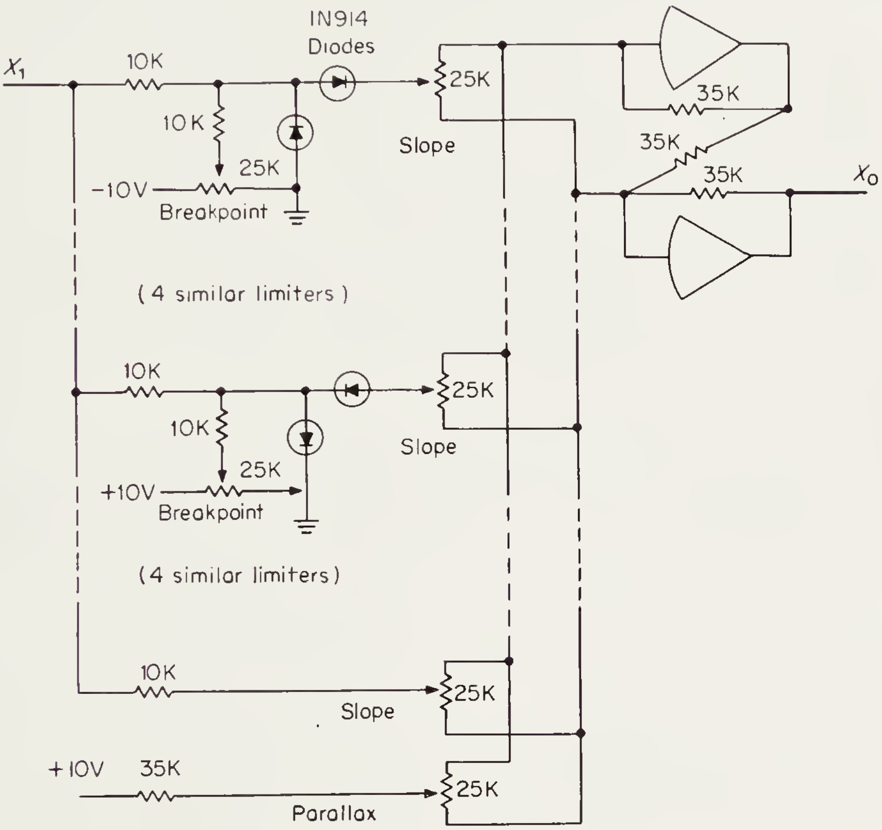


Fig. 4-35. This general-purpose diode function generator requires no quadrant switches, employs catching diodes (Sec. 4-5) for operation up to 5 KHz, and uses built-in inexpensive ±10-volt chopper-stabilized amplifiers.

Diode-function-generator specifications include:

1. *Maximum available slope per limiter channel*, typically 1 to 10 volts/volt. The total absolute function slope must be limited below 30 to 100 volts/volt to keep the summing-amplifier feedback ratio β,

and thus function-generator phase shift and noise, within reason (Sec. 3-19). A few of the limiter channels in each function generator can be given larger maximum slopes than others.

2. *Maximum error due to diode breakpoint drift*, referred to the *output;* this increases with the maximum available slope and with the number of segments and is typically between 0.05 and 0.3 volt.
3. *Amplitude, phase, and step response for small-signal sinusoidal input* with a specified function setup* and a specified resistance-capacitance load. Phase shift below 1 deg at 100 Hz and below 8 deg at 1 KHz is typical for modern designs employing series limiters. Much faster operation is possible with series-shunt limiter channels (Fig. 4-34).
4. *Output noise*, typically 30 to 300 μV rms for ±100-volt machines, depending on the maximum slope set.

Both drift and frequency response deteriorate with increasing function slope. The drift-compensation schemes of Fig. 4-12 and the use of precision limiters have replaced temperature-controlled ovens for limiter diodes.

(c) Multiple-function Generation. Some computer applications require generation of n functions $F_i(X)$ of the same machine variable X. If n exceeds 3 or 4, and if it is reasonable to use identical breakpoints for all n functions $F_i(X)$, then it may become practical to employ separate limiter amplifiers (or precision limiters) to generate precise standard-slope limiter-channel voltages. These standard segments are then summed into n sets of slope controls and summing-amplifier pairs generating each of the n functions $F_i(X)$ (Fig. 4-36).

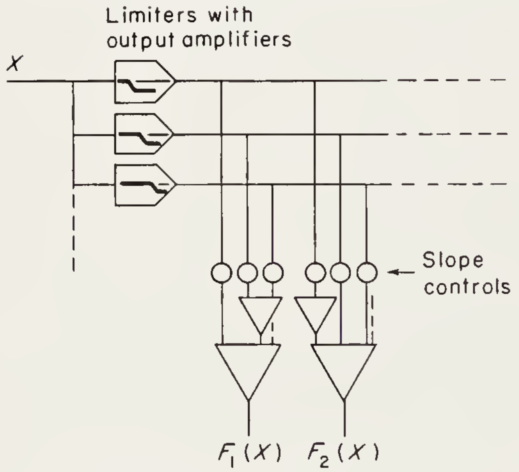


Fig. 4-36. Multiple-function generation with separate amplifier-limiters.

(d) Function Setting and Storage. A general-purpose diode function generator must permit convenient adjustment of breakpoints and slopes, with a minimum of interaction between adjustments. In view of the labor expended to set a function accurately, ease of *function storage* or *resetting* is also important.

Function setting is laborious, so that all larger analog-computer systems provide some *automatic or semiautomatic method for resetting diode function generators to stored settings:*

1. Servo-driven potentiometers or switched resistor networks controlled by punched tape, punched cards, or a digital computer

---

* A function suggested for this test is the function $X_0 = X_1$ set up with evenly spaced breakpoints and alternately canceling slopes.

2. Resistor networks switched by special function-generator patchboards
3. Storable plug-in function-generator networks

Accurate setting and resetting requires 10-turn potentiometers, or 12- to 14-bit fixed-resistor networks. Automatic resetting circuits are so expensive (and amplifiers so inexpensive) that it may pay to use precision-limiter channels in accurate function generators. Especially in ±10-volt analog computers, one may save both initial cost and setup time by supplying some limiter channels in each function generator, or some function generators in a computer, with *fixed breakpoints*. One can then combine fixed-breakpoint function generators with variable-breakpoint units, as required.

To set a given function $X_0 = F(X_1)$, one first determines suitable breakpoints and slopes by fitting a graph of $F(X_1)$ (or of an appropriate correction function, Sec. 1-12) with the desired number of straight-line segments. Analytical methods for optimum breakpoint selection[34–36] appear to be justified, if at all, only for special-purpose function generators. Assume now that we have a function generator of the type shown in Fig. 4-34*a* and require n positive breakpoints $X_1 = e_1 < e_2 < e_3 \ldots$ and n' negative breakpoints $X_1 = e'_1 > e'_2 > e'_3 \ldots$. We minimize breakpoint-drift accumulation and interaction between settings by employing *only limiter-channel segments which slope outward from zero* (Fig. 4-34*b*). We read the function-generator input X_1 and the output X_0 accurately with the aid of a digital voltmeter, a comparison potentiometer, or a special decade-switch voltage divider and employ the following procedure:

1. Set n quadrant switches for slopes to the right, and set the corresponding breakpoints all the way positive. Set the remaining quadrant switches for slopes to the left, and set the corresponding breakpoints all the way negative. Set or switch all slope controls to about one-half maximum slope.
2. With $X_1 = 0$, set the "parallax" voltage b to obtain $X_0 = F(0)$.
3. With $X_1 = e_1 > 0$, set the linear-approximation slope a to obtain $X_0 = F(e_1)$. Now adjust the e_1 breakpoint control until X_0 barely changes.
4. *Repeat steps* 2 *and* 3.
5. With $X_1 = e_2 > e_1$, set the e_1 slope to obtain $X_0 = F(e_2)$. Adjust the e_2 breakpoint control until X_0 barely changes.
6. *Repeat steps* 3 *and* 5.

We proceed outward from zero in this manner and then repeat the procedure for the negative breakpoints, again proceeding outward from zero. Finally, all breakpoints and slope settings are rechecked. An alternative setup method employs an accurate *xy* recorder (servo table) to match a desired curve as X_1 is varied.

To adjust a function generator precisely for simple analytic functions $F(X_1)$ like aX_1^2, $aX_1^2 + bX_1 + c$, ae^{bX_1}, and $a \sin bX_1$, we may be able to generate $F(\tau/\alpha_t)$ conveniently and precisely with a differential-analyzer setup (Fig. 1-4*d*) and employ the error-nulling method shown in Fig. 4-37. This method is particularly applicable to the adjustment of diode squaring networks.

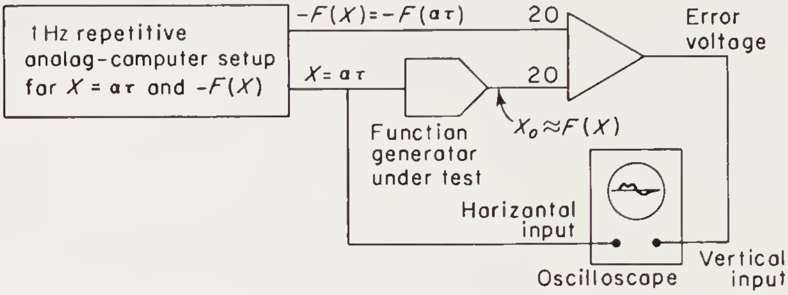


Fig. 4-37. Error-display calibration of function generators for various analytic functions such as aX^2, ae^{-bX}, and $a \sin bX$ (see also Fig. 1-4).

4-16. Logarithm-function Generators.[49-55] General-purpose analog computers are often furnished with "fixed" diode-function-generator networks producing the amplifier output $-50 \log_{10} X_1$ (voltages in percent machine units, e.g., -100 volts output for 100 volts input; see also Fig. 4-32*b*).

For less accurate logarithm (or exponential) generation in special-purpose computers, several possible schemes utilize the exponential voltage-current transfer characteristics of silicon junctions in diodes or transistors. The most generally useful circuit is the *Paterson transdiode circuit*[50] of Fig. 4-38*a*. The feedback transistor biases itself off as the output voltage nears zero. At constant temperature, the output voltage becomes

$$X_0 = -(a \log_{10} X_1 + b) \tag{4-5}$$

within 1 (true) percent over 7 to 9 decades (0.1 mV to 100 volts; refer to manufacturer's specifications for packaged log circuits in each case). With the circuit values of Fig. 4-38*a*, one has, with suitable bias adjustments,

$$a = 0.06 \text{ volt} \qquad b = 0.45 \text{ volt} \qquad (25 \text{ deg C})$$

For small signals, the transdiode acts like an (incremental) feedback transfer impedance R_0 which varies with the input current i_1 ($\approx X_1/R_1$) into the summing junction, where

$$R_0 \approx \frac{kT}{q}\frac{1}{i_1} \qquad \left(\approx \frac{1}{40 i_1} \quad \text{at 25 deg C}\right) \tag{4-6}$$

(k, T, and q are, respectively, Boltzmann's constant, absolute temperature, and electron charge). A resistance having the value (6) can be substituted for the transdiode while the amplifier is balanced for zero offset.

Because of the variable feedback, the transdiode circuit tends to become unstable for larger values of X_1 without high-frequency equalization. The relatively large feedback capacitor indicated in Fig. 4-38*a* produces stability but restricts operation to frequencies below a few hertz, as in d-c instruments and slow control circuits. The *RC* compensation circuit of Fig. 4-38*b* automatically changes the high-frequency equalization with the transistor gain and permits operation up to several kilohertz.

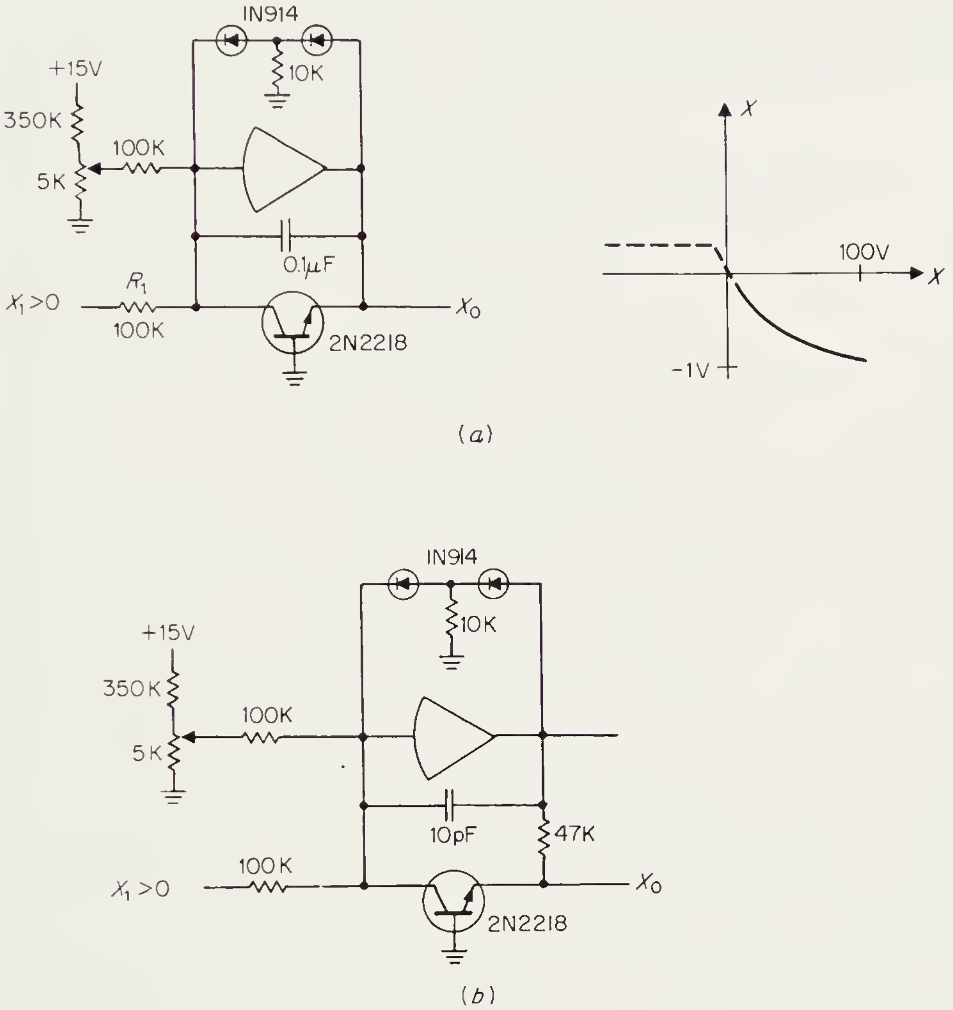


Fig. 4-38. Transdiode logarithm-generator circuit (*a*) and transdiode with automatic high-frequency equalization (*b*). Diode-limiter circuits protect each function generator against reverse input voltages. A feedback resistor equal to a typical value of $R_1/40X_1$ can be substituted for the transdiode while the amplifier is balanced. *pnp* transistors and reversed diodes similarly yield reversed logarithmic characteristics.

Unfortunately, the scale factor a in Eq. (5) increases with temperature at the rate of 0.3 percent/deg C. Numerous temperature-compensation methods involving either an extra diode or transistor or a thermistor have been proposed.[99] Perhaps the best scheme is to compute the logarithm of a voltage *ratio*, or $\log_e X_1/X_2$ with matched transdiode transistors in the circuit of Fig. 4-39*a*. The temperature dependence will cancel approximately to reduce the temperature coefficient of the output by at least a factor of 10, and the second input voltage X_2 can be a constant if desired. The term b in Eq. (5) also changes with temperature; this adds an offset voltage of 100 to 300 μV/deg C *referred to the amplifier input* (Sec. 3-15) to the temperature-dependent offset of the amplifier proper.

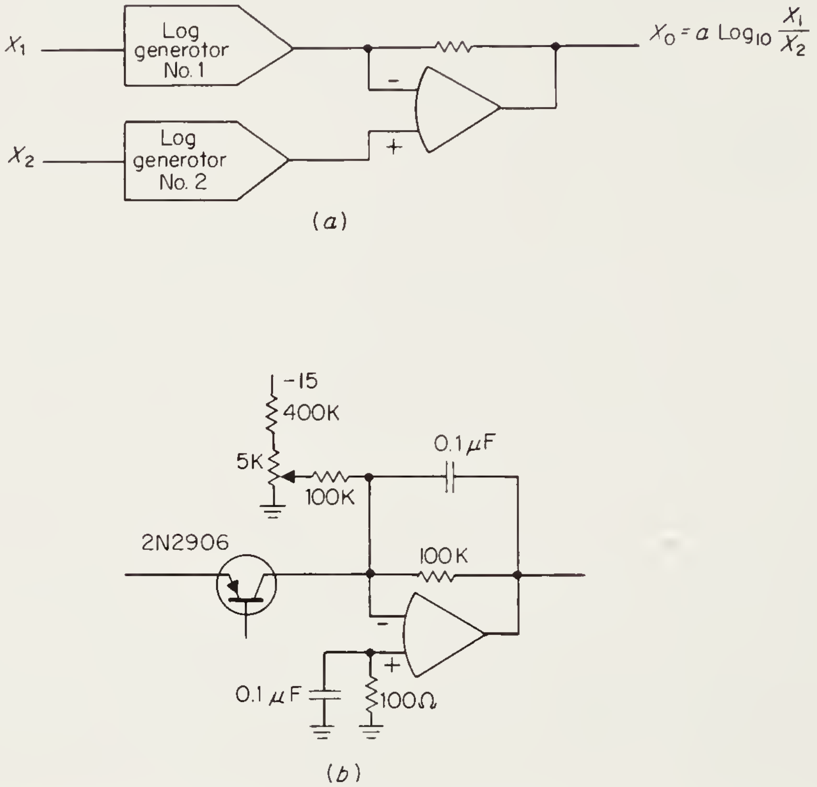


Fig. 4-39. A log-ratio circuit used for temperature compensation or division (*a*) and a circuit for generating an exponential (antilog) characteristic (*b*). This circuit can be used in connection with two logarithm generators to produce products of the form $X^a Y^b$. (*Based on Ref.* 52.)

Since amplifier bias-current changes in Fig. 4-38 will affect the term b in Eq. (5), the amplifiers used should have low input current (FET or super-beta input stages, Sec. 3-33).

4-17. Miscellaneous Circuits. (a) General Remarks.

Function generators based on the nonlinear transfer characteristics of diodes or transistors (Sec. 4-16), nonlinear resistance elements (Sec. 4-17*b*), and special cathode-ray-tube circuits (Sec. 4-17*c*) are rarely more accurate than within 0.5 to 3 percent, but may be preferred to diode function generators in special situations where piecewise-linear function approximations are objectionable (e.g., where the function output must be differentiated). Reference 93 describes the construction of special nonlinear resistors by silicon-integrated-circuit processing techniques.

(b) Varistor Function Generators.[56–58] Varistor function generators (Fig. 4-40) employ current-sensitive nonlinear resistors (silicon-carbide varistors, e.g., Thyrite, Globar) to produce transfer characteristics of the general form shown in Fig. 4-40*b*. The corresponding inverse functions can be generated by the method of Fig. 4-32*a*.

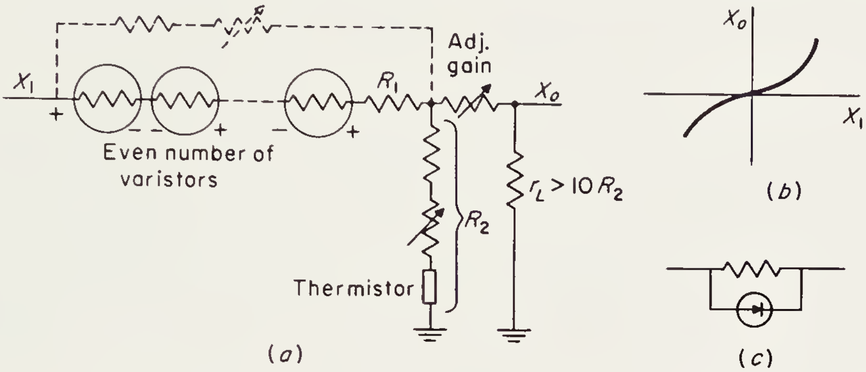


Fig. 4-40. Varistor function-generator network (*a*), transfer characteristic (*b*), and diode network used to compensate for varistor rectification (*c*). With 10 General Electric Type 839 6839 G1 varistors, $R_1 + R_2$ is between 5 and 20K. r_L can be an input resistor of an operational amplifier.

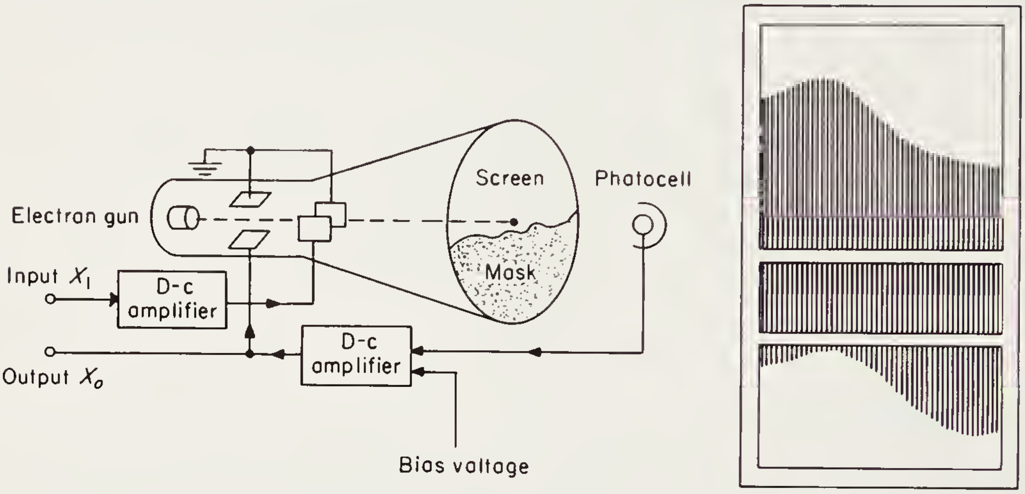
Schreier[57,58] discusses a wide variety of varistor circuits including sinusoid generators, calibration methods, and test results. Factory-adjusted varistor squaring units are commercially available (Quadratron, Bourns, Inc., Riverside, Calif.); combinations of squaring and cubing units can approximate various other functions, such as truncated power series. Varistor function generators for analytic functions are best adjusted by the error-nulling method of Fig. 4-35.

Figure 4-40*a* shows a fairly general varistor network; adjustment of the resistances R_1, R_2 yields different transfer characteristics. Series combination of between 2 and 10 individual varistors permits input voltages up to 100 volts and tends to average manufacturing tolerances. Since silicon-carbide varistors exhibit a slight rectification effect (between 0 and 1 percent), one either attempts to match series-opposed varistors, or one adds a suitably polarized diode network (Fig. 4-40*c*) in series with R_1 or R_2. A thermistor (Fig. 4-40*a*), or a small crystal oven for the entire network, will minimize the effect of varistor-resistance changes with ambient temperature.[56] Varistor self-heating is not so easily compensated and accounts for a possible drift error between 0.2 and 1 percent of half-scale added to between 0.1 and 0.8 percent error in matching desired function characteristics.

Distributed capacitance combines with the variable varistor resistance to produce phase shift which varies with the input voltage and usually limits the use of uncompensated varistor networks to frequencies below 100 Hz. This phase shift can, however, be partially compensated with a voltage-sensitive capacitor (silicon-junction diode designed for this purpose) either across R_2 or across the feedback resistor of a following operational amplifier. Another compensation scheme feeds the inverted input voltage through a trimmer capacitor around the varistors; the varistor capacitance as such does not change appreciably for currents below 1 mA.

(c) Cathode-ray-tube Function Generators. In the *photoformer* function generator of Fig. 4-41,[60–65] a feedback loop comprising the cathode-ray tube, the photocell, and the vertical deflection amplifier keeps the spot on the screen just below the edge of a prepared function mask as the input voltage X_1 (horizontal deflection voltage) varies. Hence, the output voltage X_0 (vertical deflection voltage) changes as a function $X_0 = F(X_1)$ of X_1. For convenience, masks can be provided with calibration steps (0 and ± 50 volts) to the left or right of the function curve. A comprehensive discussion of photoformer accuracy is presented in Ref. 65.

In a second type of cathode-ray-tube function generator,[64] the electron beam is positioned horizontally by the input voltage X_1, while a linear sawtooth sweep causes the beam to sweep vertically so as to scan an opaque or transparent function mask between 5,000 and 50,000 times per second. For each value of X_1, the photocell current is, then, pulse-width modulated in accordance with the function $F(X_1)$ shaping the mask. The resulting pulse-width-modulated square wave is filtered to yield $F(X_1)$ and can even be used to multiply other variables by $F(X_1)$, as in a time-division multiplier (Sec. 6-9).



(*a*) (*b*)

Fig. 4-41. Block diagram of a photoformer (*a*) and adjustable photoformer mask consisting of clamped shims (*b*).

The scanning scheme is relatively insensitive to screen and phototube persistence, changes in spot size, and variations of deflection sensitivity. Provision of an opaque or transparent reference line yields standard reference pulses for automatic calibration.

Generally speaking, diode function generators are easier and less expensive to construct than cathode-ray-tube function generators, but the latter permit convenient and rapid function changes if stored function masks are available. Useful response up to 100 KHz is possible with photoformers, and up to about 1 KHz with cathode-ray scanners. With function masks prepared directly from function data, either type of cathode-ray-tube function generator can yield a static accuracy within 1 percent of half-scale. This figure can, however, be improved if we minimize the effects of mask parallax, finite spot size, and variable deflection sensitivity:

1. By employing *adjustable function masks* (Fig. 4-41*b*) set so that the *measured* output voltage matches the desired function
2. By preparing function masks photographically *with the function generator itself*, using known deflection voltages

It is, of course, again preferable to use the function generator only for correcting a linear approximation.

FUNCTIONS OF TWO OR MORE VARIABLES

4-18. Introduction. Interpolation Schemes. Although many functions $F(X, Y, \ldots)$ of several machine variables X, $Y, \ldots$ can be generated by the usual combinations of summers, multipliers, and single-variable function generators (Fig. 1-4), many simulation problems require generation of empirically tabulated or charted functions of two of more variables (e.g., vacuum-tube and transistor characteristics, drag as a function of airspeed and angle of attack), and similar requirements arise in instrumentation and control applications. Note that combinations of two-variable function generators can generate functions of *any* finite number of variables; thus, $F(X,Y,Z) = G[f(X,Y),Z]$.

Unless a convenient analytical approximation applies, the sheer amount of data involved makes the setup of most two-variable functions $F(X, Y)$ quite laborious, no matter what type of function generator is used. In applications requiring frequent changes of such functions, some sort of a hybrid analog-digital function generator (Sec. 9-9), or at least a digitally controlled setup system (Sec. 7-4*b*), will pay for itself. As regards accuracy, the reader is reminded once again that the best way to use any general-purpose function generator may be to generate only *corrections* to an analytical approximation of the desired function.

To generate an empirically determined function $F(X, Y)$ with conventional computing elements, $F(X, Y)$ must be fitted with an approximation function whose choice will depend on the particular application. A frequently reasonable approach is to generate $F(X, Y_i)$ accurately for a number of fixed values $Y = Y_1, Y_2, \ldots$ and to *interpolate* for other values of Y. Since sufficiently accurate linear interpolation might require too many function generators for the $F(X, Y_i)$, one can use a nonlinear interpolation formula like

$$F(X,Y) = f_1(X) + (Y - Y_1)f_2(X) + (Y - Y_1)(Y - Y_2)f_3(X) + \cdots \tag{4-7}$$

Three terms are often sufficient. The functions $f_1(X)$, $f_2(X), \ldots$ are computed by successive substitution of $Y = Y_1, Y_2, \ldots$ in Eq. (7).

Figure 4-42 shows two *mechanical interpolation schemes* adaptable for use with conventional analog computers. The Link Division of General Precision, Inc.,* has also developed a two-dimensional analog of the well-known tapped potentiometer. A sheet of resistive material placed on a small servo table bears a grid of point contacts set to the desired voltages $F(X_i, Y_k)$ by padding-resistor networks, and a servo-positioned probe moves over the paper to provide two-dimensional interpolation.

---

* Binghamton, N.Y.

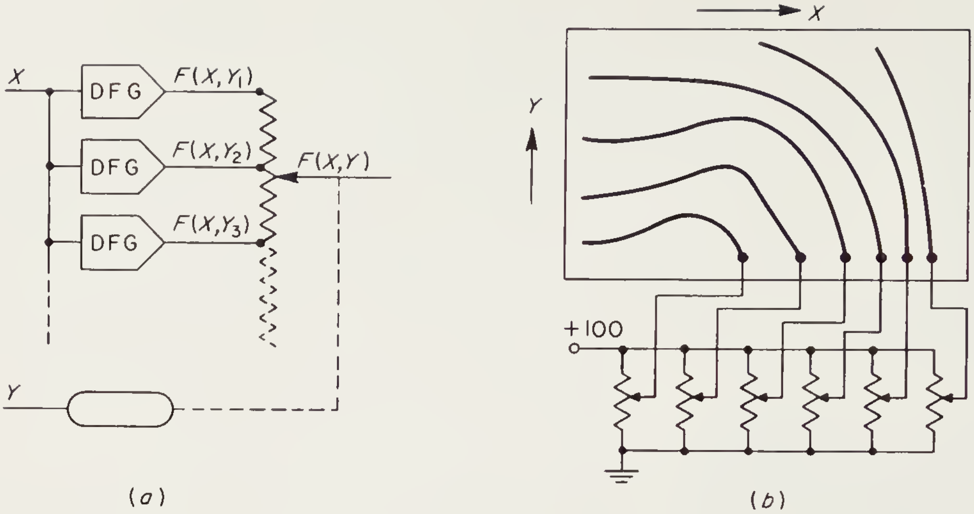


Fig. 4-42. Interpolation with a tapped potentiometer (*a*), and use of a servo plotting table with resistive paper for interpolation between contour lines $F(X,Y) = \text{const}$ drawn with conducting ink. In Fig. 4-42*b*, an ordinary *xy* recorder positions a probe sensing $F(X,Y)$; frequency response is limited to below 3 Hz for ½-in. excursions. Function sheets can be stored for reuse.

4-19. Diode Function Generators. (a) Simple Limiter Circuits. Very useful piecewise-linear approximations to many functions of several variables can be generated by simple limiter circuits (Figs. 4-2, 4-4, 4-5, 4-9, etc.) *with variable bias voltages.* As a useful example, Fig. 4-43*a* shows one of many possible methods for approximating transistor characteristics with simple limiters (see also Ref. 75).

(b) Use of General-purpose Diode Function Generators. More general types of functions $F(X,Y)$ can be generated with ordinary multipurpose diode function generators if the breakpoint-bias voltages are varied as functions of the second input variable Y. If, for instance, a variable input Y replaces the constant reference voltage E in the diode

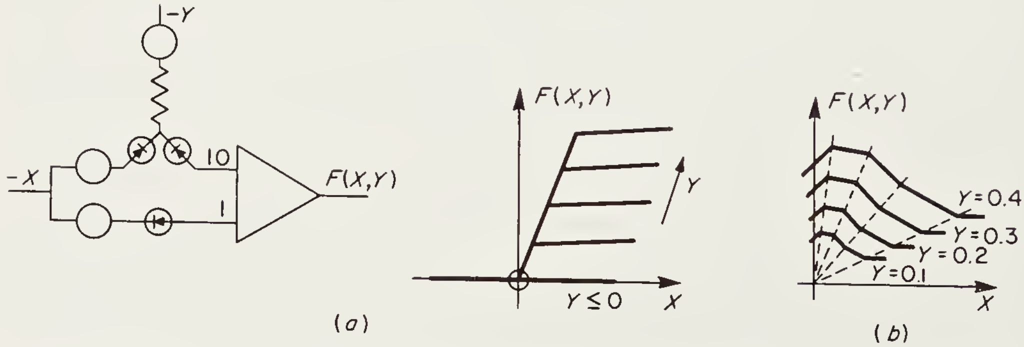


Fig. 4-43. Example of a piecewise-linear function of two variables generated with patchbay-connected diode limiters (*a*), and more general function $F(X,Y)$ generated by the diode function generator of Fig. 4-34 with variable bias $E = Y$ (*b*).

function generator of Fig. 4-34, the output voltage X_0 will be a polygonal function of X for each value of Y (Fig. 4-43*b*), so that a function $F(X,Y)$ is produced.[37]

Since the breakpoint voltage of the ith limiter channel will be $b_i Y$, where b_i is a constant, all breakpoints generated by any one limiter channel must lie on a straight line through the origin. These *breakpoint loci* bound segments of equal slope and are indicated in broken lines in Fig. 4-43*b*.

One can further generalize the above procedure by adding a function of X or Y to the output voltage X_0. One obtains an even less restricted class of functions $F(X,Y)$ by introducing suitable *functions* $b_iG(Y)$ or $G_i(Y)$ of Y as bias voltages in the limiter channels of Fig. 4-34, so that the breakpoint loci are no longer straight lines. To set up a given function $F(X,Y)$, one must first locate the breakpoint loci by joining points of equal slope on a set of parametric curves like those in Fig. 4-43*b*.

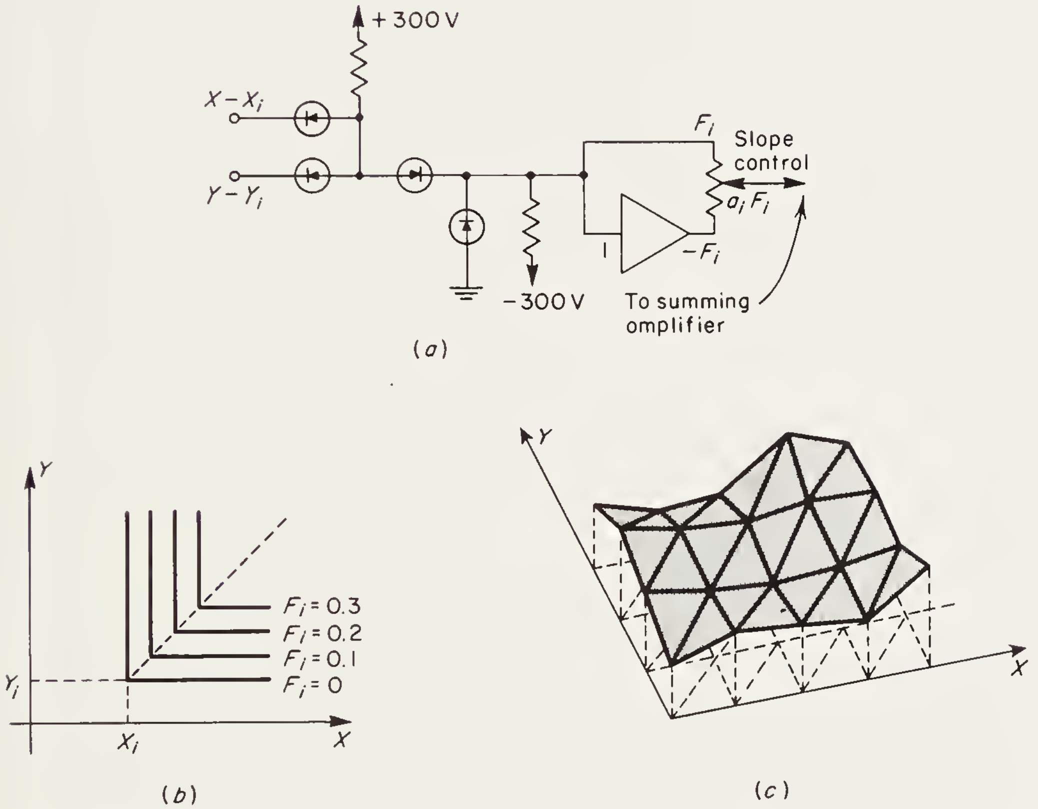


Fig. 4-44. Generation of

$$F_i(X,Y) = \max\,[\min\,(X - X_i,\ Y - Y_i), 0]$$

by a double selection circuit (*a*), contour lines of $F_i(X,Y)$ (*b*), and approximation of a function $F(X,Y)$ by

$$F(X,Y) \approx \sum a_i F_i(X,Y)$$

Note the triangular facets.

(c) Special Diode Function Generators. The Philbrick Model F2V function generator (G. A. Philbrick Researches, Inc., Boston, Mass.)[77] employed a set of amplitude-selector channels (Fig. 4-44*a*) to produce a desired function $F(X,Y)$ as a weighted sum of functions

$$F_i(X,Y) = \max\,[\min\,(X - X_i,\; Y - Y_i),0] \qquad (4\text{-}8)$$

The function (8) may be regarded as one two-dimensional generalization of the line segments generated in ordinary diode function generators. The contour lines of Fig. 4-44*b* illustrate the nature of a typical channel output $F_i(X,Y)$. To set up a desired function $F(X,Y)$, one first chooses the breakpoints (X_i,Y_i) and adjusts the slope controls to obtain the desired function values $F(X_i,Y_i)$ for successively increasing values of X_i and Y_i. Figure 4-44*c* shows a three-dimensional graph of a function $F(X, Y)$ generated by a Philbrick-type function generator.

A possibly simpler way to produce, say, a downward-convex function $F(X, Y)$ is to generate a set of voltages $Z_i = a_iX + b_iY + c_i$ which represent planes whose segments are to approximate the desired surface. An amplitude-selector circuit (Fig. 4-5 or 4-25) is then used to obtain $\max\,(Z_1,Z_2, \ldots)$. More complicated surfaces such as hyperboloids will require two such selection processes.[81] References 87 to 90 describe the mathematical theory (a generalization of Boolean algebra) needed for the systematic design of such function generators, which may be of considerable interest for special-purpose computing devices. In general-purpose analog/hybrid computers, setup convenience has led to almost universal acceptance of hybrid analog-digital function generation for functions of two or more variables (Secs. 9-9 and 9-10).

REFERENCES AND BIBLIOGRAPHY

References 1 to 3 are primary references on device models. See also the current application bulletins published by manufacturers of semiconductors and of discrete and integrated-circuit operational amplifiers.

General References

1. Millman, J., and H. Taub: *Pulse, Digital, and Switching Waveforms*, McGraw-Hill, New York, 1965.
2. Gray, P. E., et al.: *Physical Electronics and Circuit Models of Transistors*, Wiley, New York, 1965.
3. Phillips, A. B.: *Transistor Engineering*, McGraw-Hill, New York, 1962.
4. *Applications Manual*, Philbrick/Nexus Research, Dedham, Mass., 1966.
5. *Catalog and Handbook*, Burr-Brown Research Corporation, Tucson, Ariz., 1971.

Limiter Circuits

6. Morrill, C. D., and R. V. Baum: Diode Limiters Simulate Mechanical Phenomena, *Electronics*, November 1952.
7. Pfeiffer, P. E.: A Four-quadrant Multiplier, *IRETEC*, June 1959.
8. Howe, R. M.: The Use of Silicon-junction Diodes in Analog Simulation, *Instr. Control Systems*, August 1961.

9. Schmid, H.: Transistor Analog Computing Elements, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
10. Medkeff, R. J., and R. J. Parent: A Diode-bridge Limiter for Use with Electronic Analog Computers, *Proc. AIEE*, vol. 70, sec. T-1-170, 1951.

Comparators

11. Kley, A., et al.: Schnelle und genaue Komparatoren, *Int. Elektron. Rundschau*, January 1967.
12. Cleverley, D. S.: Tunnel-diode Threshold Detector, *Electron. Design*, Sept. 13, 1963.
13. Moulton, A. B.: Tunnel-diode/Transistor Comparators, *Electron. Design*, Aug. 17, 1964.
14. Mott, H., and C. C. Carroll: A Versatile Comparator, *IEEETEC*, May 1965.
15. Agarwal, V. K.: Tunnel Diodes Speed Up Comparator, *EDN*, July 15, 1969.

Precision Limiters and Related Circuits

16. Morrill, C. D., and R. V. Baum: Diode Limiters Simulate Mechanical Phenomena, *Electronics*, November 1952.
17. Koerner, H., and G. A. Korn: Function Generation with Operational Amplifiers, *Electronics*, Nov. 6, 1959.
18. Korn, G. A.: Control Applications for New Deadspace Limiter, *Control Eng.*, March 1962.
19. ———: A Multi-comparator Circuit, *Instr. Control Systems*, January 1963; see also *Ann. AICA*, April 1963.
20. ———: Maxima and Minima, in T. C. Anderson, Analog Techniques, *Instr. Control Systems*, March 1962.

Bistable Operational-amplifier Circuits and Special Signal Generators

21. Morrill, C. D., and R. V. Baum: Diode Limiters Simulate Mechanical Phenomena, *Electronics*, November 1952.
22. Eckes, H. R.: A New Analog-simulated Bistable Element Permitting Symmetrical Triggering, *IRETEC*, October 1962.
23. Howe, R. M., and C. Leite: Low-frequency Oscillator, *Rev. Sci. Instr.*, **24:** 901 (1953).
24. Cochran, D. S.: The Transistorized RC Oscillator, *Hewlett-Packard J.*, Hewlett-Packard Co., Palo Alto, Calif., January 1962.
25. Oliver, B. M.: The Effect of μ-circuit Nonlinearity on the Amplitude Stability of RC Oscillators, *Hewlett-Packard J.*, Hewlett-Packard Co., Palo Alto, Calif., April 1960.
26. Witsenhausen, H.: Three-phase Oscillator, *Ann. AICA*, April 1962.
27. Meyer-Brötz, G.: Anwendungen analoger Rechenelemente in der Tiefstfrequenz-Messtechnik, *Frequenz*, **16:** 1 (1962).
28. Klein, G., and J. M. Hertog: A Sine-wave Generator with Periods of Hours, *Electronic Eng.*, June 1959.
29. Conant, B. K.: Analog-computer Setup for an Accurate Monostable Multivibrator, *Instr. Control Systems*, June 1963.
30. Hausner, A.: Multiple Integrals on a Non-repetitive Analog Computer, *Proc. Spring Joint Computer Conf.*, Detroit, Mich., 1963.
31. Foote, D. D.: Pulse-width Modulator, *Instr. Control Systems*, May 1963.
32. Panico, J. J.: FET Stabilizes Amplitude of Wien-bridge Oscillator, *Electronics*, Oct. 3, 1966.
33. Pawluk, H.: FET Levels Wien-bridge Oscillator Output, *Electron. Eng.*, October 1967

Piecewise-linear Function Approximations

34. Goldhirsh, I. L.: Computation of Continuous Functions, *Simulation*, July 1968.
35. VanSteenkiste, G. J. V.: Optimal Generation of Arbitrary Functions, *Ann. AICA*, July 1967 (see also review by E. J. Copes in *IEEETEC*, July 1968).
36. Gracon, T. J., and J. C. Strauss: Application of Dynamic Programming to Function Generation, *Simulation*, November 1968.

Diode Function Generators

(See also data sheets on commercially available computers)

37. Meissinger, H. F.: An Electronic Circuit for the Generation of Functions of Several Variables, *IRE Natl. Convention Record*, 1955.
38. Miura, T., et al.: A New Diode Function Generator, *IRETEC*, June 1957.
39. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill, New York, 1959.
40. Fitsner, L. N., and I. Doganovskii: On a Method for Computing the Electrical Network for Nonlinear Diode Generators, *Priborostroenie*, January 1959; *Automation Express*, April 1959.
41. Maslov A. A., and Y. G. Purlov: A Universal Function Generator Based on the Principle of Quadratic Approximations, *Automatika i Telemekhanika*, **21** (February 1960).
42. Mayer, R. E., et al.: Apparatus for Setting a Function Generator, U.S. Patent 2,964,709, Dec. 13, 1960 (filed Dec. 31, 1957), assigned to Reeves Instrument Corp., Garden City, N.Y.

Temperature Compensation of Diode Limiters and Function Generators

43. Schneider, W.: Fehlerabschatzung bei Diodenfunktionsgebern, *Telefunken Z.*, January 1966.
44. Kley, A., and E. Heim: Funktionsgeber und Multiplizierer mit Halbleiterdioden, *Telefunken Z.*, January 1966.
45. ———: Ein elektronischer Koordinatenwandler, *Telefunken Z.*, January 1966.
46. Whigham, R. H.: A Fast 10-volt Diode Quarter-square Multiplier, *Simulation*, August 1965.
47. Giloi, W., and R. Fritz: Computer-aided Design of Quarter-square Multipliers, *Simulation*, September 1968.
48. Hoag, D. M.: Op Amp Clipper Design Offers Thermal Stability, *Electro-Technology*, January 1970.

Transistors and Diodes as Logarithmic Elements

49. Transistor Function Generator Computes Logarithms, *Electron. Design*, Aug. 2, 1963.
50. Paterson, W. L.: Multiplication and Logarithmic Conversion by Operational-amplifier/Transistor Circuits, *RSI*, December 1963.
51. Gibbons, J. F., and H. S. Horn: A circuit with Logarithmic Transfer Response over Nine Decades, *IEEE Trans. Circuit Theory*, September 1964.
52. Byrd, J. S.: Eight-decade Logarithmic Converter, *ISAJ.*, August 1966.
53. Transistor Logarithmic Conversion, *Application Note* 261, Motorola Semiconductor Products, Inc., Phoenix, Ariz., 1968.
54. Walker, R. E.: Log Cards to Raise a Variable to a Power, *Simulation*, January 1968.
55. Gurski, R. J.: Logarithmic Devices, *Lightning Empiricist*, Philbrick-Nexus Research, Dedham, Mass., March 1969.

Varistor Function Generators

56. Kovach, L. D., and W. Comley: Nonlinear Transfer Functions Using Thyrite, *IRETEC*, June 1958; see also A New Solid-state Analog Component, *IRETEC*, December 1960.
57. Schreier, D.: Darstellung nichtlinearer Characteristiken unter Verwendung von Varistoren, *Wiss. Z. Tech. Hochsch. Ilmenau*, 7 (2) (1961).
58. ———, and H. Winkler: Design of Nonlinear Characteristics Using Varistors, *Proc. AICA 3d Intern. Conf.*, Opatija, Yugoslavia, September 1961; Presses Académiques Européennes, Brussels.
59. Bloom, H. M.: Circuit Parameters for Optimum Quadratron Performance, TM-66-15, Harry Diamond Laboratories, Washington, D.C., 1966.

Cathode-ray Function Generators

60. Mynall, D. J.: Electronic Analogue Computing, *Electron. Eng.*, August 1947.
61. Sunstein, D. E.: Photoelectric Waveform Generator, *Electronics*, February 1949.

62. Hancock, E. J.: Photoformer Design and Performance, *Proc. Natl. Electron. Conf.*, **7**: 288 (1951).
63. Gerlach, A. A., C. N. Pederson, and R. E. Zenner: A Precise Electronic Function Generator, *Proc. Natl. Electron. Conf.*, **8** (1952).
64. Polimerou, L. G.: A New Method of Generating Functions, *IRETEC*, September 1954.
65. Miura, T., and R. Abe: Error Analysis of a Photoformer, *Electron. Tech. J. Japan*, **5** (1) (1959).

Tapped-potentiometer Function Generators

66. Korn, G. A.: Design and Construction of Universal Function Generating Potentiometers, *Rev. Sci. Instr.*, **21**:77 (1950).
67. Harris, H. E., and R. D. McCoy: Simplified Calculations for Servo Function Generators, *Electron. Design*, March 1957.
68. Williams, R. W., and H. Marchant: Resistance Potentiometers as Function Generators, *Electron. Eng.*, October 1958.
69. Garner, K. C.: Linear Multi-tapped Potentiometers with Loaded Outputs, *Electron. Eng.*, April 1959.
70. Kislitsyn, S. G., and F. L. Litvin: On Calculating Function-generating Shunted Potentiometers for Chebyshev Approximations, *Automatika i Telemekhanika*, November 1959.
71. Sherchaut, I.: The Correction of Errors in Potentiometer Function Generators, *Electron. Eng.*, August 1960.
72. Parezanovic, N., and M. Dujmovic: Accuracy Improvements of the Tapped-potentiometer Function Generators, *IRETEC*, February 1962.
73. Korn, G. A.: Calibration of Padded-potentiometer Function Generators, *Instr. Control Systems*, October 1959.

Functions of Two or More Variables; Use of Maximum/Minimum Selector Logic

74. Larrowe, V., and M. Spencer: Use of Semi-conducting Surfaces in Analog Function Generation, *Eng. Rept.*, Willow Run Research Laboratory, University of Michigan, Ypsilanti, Mich., 1959.
75. Rosenkranius, L.: Analog-computer Techniques for the Prediction of Transient Nuclear Radiation Effects on Transistor Circuits, *AIEE Conf. Paper* CP 62-1082, Summer General Meeting, Denver, Colo., 1962.
76. Meissinger, H. F.: An Electronic Circuit for the Generation of Functions of Several Variables, *IRE Natl. Convention Record*, 1955.
77. *Data Sheet for Model* F2V *Function of Two Variables Component*, G. A. Philbrick Researches, Inc., Boston, Mass., November 1957.
78. Variable Density Film Yields Three Functions, *Control Eng.*, May 1956.
79. Polimerou, L. G.: A New Method for Generating a Function of Two Independent Variables, *IRETEC*, September 1957.
80. Wentzel, V.: Two Electronic Function Generators, *Proc. 1st AICA Intern. Conf.*, Brussels, 1955, Presses Académiques Européennes, Brussels, 1956.
81. Petternella, M., and A. Ruberti: A Diode Generator of Functions of Two Variables, *Ann. AICA*, January 1961.
82. Smolov, V. B.: Two-input Electronic Function Generators, *Automatika i Telemekhanika*, October 1959.
83. Peretz, R.: Calculateur I.R.S.I.A.-TABLELEC, *Proc. 2d AICA Conf.*, Strasbourg, France, 1958, Presses Académiques Européennes, Brussels, 1959.
84. ———: Opérateurs analogiques spéciaux, *Proc. 2d AICA Conf.*, Strasbourg, France, 1958, Presses Académiques Européennes, Brussels, 1959.
85. Kelley, T.: Multi-variable Function Generator, *Instr. Control Systems*, November 1959.
86. Visman, A. C. P.: A Diode Function Generator Made Suitable for Two Variables, *Ann. AICA*, January 1965.
87. Stern, T. E.: Piecewise-linear Network Theory, *MIT Res. Lab. Electron. Rept.* 315, Cambridge, Mass., June 15, 1956.

88. Wilkinson R. H.: A Method of Generating Functions of Several Variables Using Analog Diode Logic, *IEEETEC*, April 1963; reprinted in *Simulation*, May 1964.
89. Ginzburg, S. A.: Continuous Logic and Function Generation, *Proc. 5th AICA Conf.*, Lausanne, Switzerland, 1967, Presses Académiques Européennes, Brussels.
90. ———: *Mathematical Continuous Logic and the Representation of Functions*, Moscow, 1968 (in Russian); edited machine translation FTD-MT-24-137-68 (TM 850 1150), Foreign Technology Division, Wright-Patterson Air Force Base, Ohio, 1968.

Miscellaneous Circuits

91. Paynter, H. M.: Ordering and Selection Processes and Ultrareliable Systems, AAAS paper, December 1958; reprinted by G. A. Philbrick Researches, Inc., Boston, Mass.
92. Brunner, W., and R. Beadle: Amplitude Stabilization of Harmonic Oscillator with Diode Gates, *PCC Rept.* 173, Electronic Associates, Inc., Princeton, N.J., 1962.
93. Post, P. B.: Silicon Chips Generate Analog Computing Functions, *Electro-Technology*, May 1968.
94. Conant, B. K.: A New Solid-state Iterative Differential Analyzer, *Proc. FJCC*, 1968.
95. Stata, R.: *Application Notes*, Analog Devices, Inc., Cambridge, Mass.; see also *Electromechan. Design*, September and November 1954.
96. Caldwell, W. F., G. A. Korn, et al.: A Precision Amplitude-distribution Analyzer, *IRETEC*, June 1952.
97. Brubaker, T. A., and G. A. Korn: Accurate Amplitude Distribution Analyzer, *Rev. Sci. Instr.*, March 1961.
98. Widlar, R. J.: Fast Voltage Comparators with Low Input Current, *Application Note* LB-6, National Semiconductor Corp., Santa Clara, Calif., 1968.
99. ———: IC Op Amp Beats FETs on Input Current, *EEE*, December 1969; *Application Note* AN-29, National Semiconductor Corporation, Santa Clara, Calif.

Hybrid Analog-digital Function Generation

(See Chap. 9)

CHAPTER 5

SWITCHING, CONVERSION, SAMPLING, AND MODE CONTROL

ANALOG SWITCHING CIRCUITS

5-1. Applications and Desirable Specifications. **(a)** *Analog switches* open and close analog-current connections in accordance with the discrete states of "digital" circuits or commands (1 for ON, 0 for OFF). *They are, therefore, the basic digital-to-analog conversion devices.* Accurate analog switches are required for:

1. General *programming* of analog circuits
2. *Time sharing* (*multiplexing and demultiplexing*) of scarce computing, conversion, or communication devices
3. *Digital-to-analog conversion*
4. *Sampling* of analog signals
5. *Integrator mode control*
6. *Modulation and demodulation* (e.g., chopper circuits, Sec. 3-36)

(b) An *ideal switch* closes a desired circuit with *zero forward resistance* and opens it with *infinite back resistance;* the ideal switch has *no spurious current or voltage offsets* and opens and closes *instantaneously.* Moreover, the digital control signal of an ideal analog switch will not affect the analog output signal (perfect *isolation* of control and output). The static characteristics of ideal switches are accurately approximated by mechanical switches and relays with precious-metal contacts. When fast analog or hybrid analog-digital computation requires switching times less than (at best) 1 msec, mechanical switching causes intolerable time-delay errors (Sec. 3-2*c*). Relays also create maintenance problems. *Electronic switches* permit switching times between 10 and 2,000 nsec, but will have finite forward and back resistances and produce spurious offset voltages and currents; accurate switches may require separate d-c balance controls for the ON and OFF conditions. Again, proper electronic-switch design must minimize interference of the abrupt control-voltage steps with the switched signal. The best electronic switching circuits employ ingenious *combinations* of electronic switches or of *electronic switches and operational amplifiers.*

(c) An electronic switch can be inserted *in series with the signal current* (Fig. 5-1*a*), or it can *shunt the signal to ground* (Fig. 5-1*b*). At higher signal frequencies, and in most critical applications, *combined shunt and series switching* is recommended to reduce capacitive as well as resistive leakage (Fig. 5-1*c*, *d*).

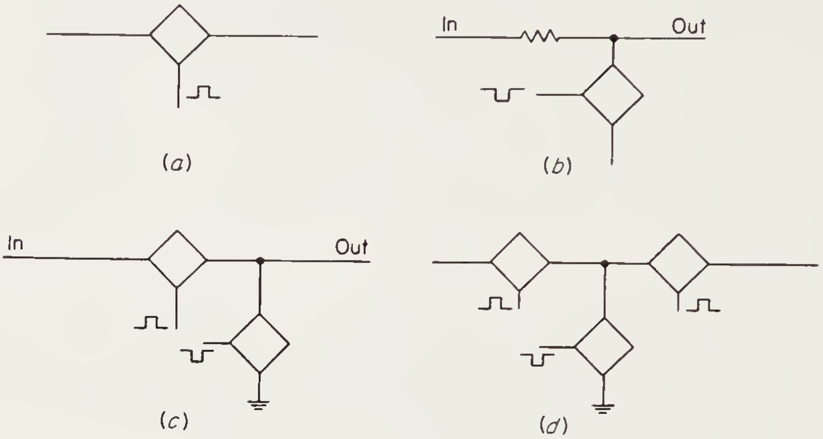


Fig. 5-1. Electronic series switch (*a*), shunt switch (*b*), and combined shunt-series switches (*c*), (*d*).

If three identical switches with back resistance R_B and open-circuit capacitance C_B are used in the circuit of Fig. 5-1*d*, its short-circuit transfer impedance (Sec. 1-10) in the OFF condition is

$$Z = \frac{2R_B}{j\omega R_B C_B + 1} + \frac{R_B^2}{R_F(j\omega R_B C_B + 1)^2} \tag{5-1}$$

where C_B is the leakage capacitance of each open switch. We see that capacitive as well as resistive leakage is shunted by R_F. For $R_F = 50$ ohms, $R_B = 50$M, $C_B = 5$ pF, $|Z|$ exceeds 5×10^7M at d-c, 2,000M at 100 KHz, and 20M at 1 MHz.

(d) Most electronic switches are essentially single-throw (ON or OFF) switches. Figure 5-2 indicates how such electronic switches can, nevertheless, implement double-throw action and polarity reversal.

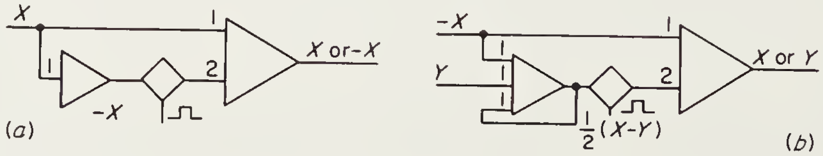


Fig. 5-2. Simulation of an SPDT switch with an SPST electronic switch.

(e) *Electronic-switch control voltages (usually steep-sided voltage steps from digital logic) can cause two types of error in the analog signal circuit:*

1. If control and signal currents run through an appreciable common impedance (signal source and/or load, or switch impedance), then the switched signal will contain spurious step voltages ("pedestals").
2. Switch capacitances between control and signal circuits cause *switching spikes* in resistive signal circuits as the turn-on and turn-off steps are differentiated (Fig. 5-3). With capacitive signal loads, the charges transmitted through the spurious coupling capacitances cause step errors (Sec. 5-17).

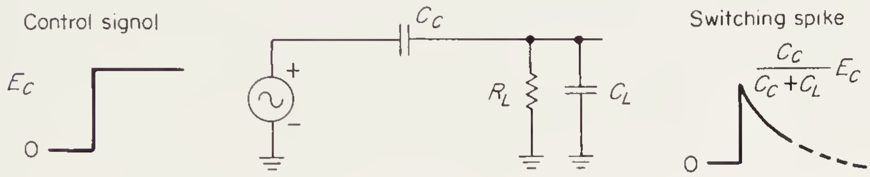


Fig. 5-3. Turn-on spike (grounded control-signal source, grounded load) due to a switch capacitance C_C. The dashed lines indicate that the spike will be modified when the switch actually turns on. Until then (neglecting nonlinear-circuit effects), the spike has the form of a decaying exponential with initial amplitude $E_C C_C/(C_C + C_L)$ and time constant $R_L(C_C + C_L)$. For a purely capacitive load ($R_L = \infty$), the exponential becomes a step.

Most control logic is grounded on one side. Unless it is possible to use "floating" (transformer- or light-coupled) control circuits (Sec. 5-3), one must minimize pedestal errors by connecting one end of each electronic switch to ground (as in shunt switching) or to a low-impedance source (series switching).

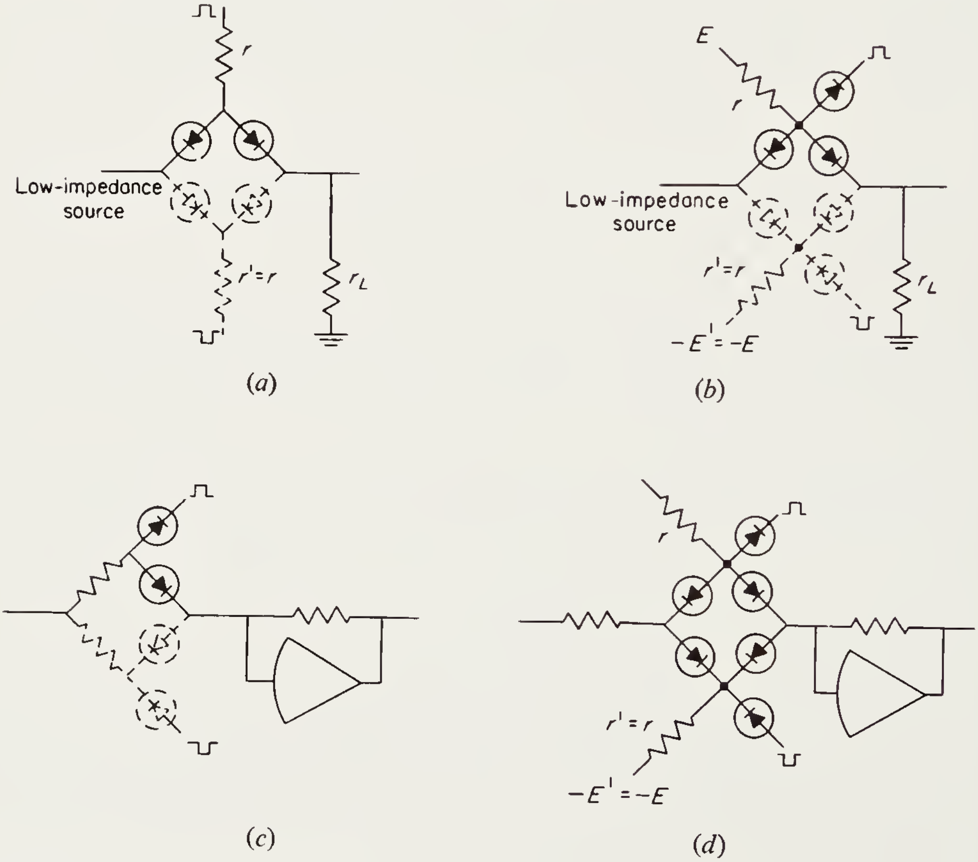
Switching spikes are not easily "grounded out" in this manner (1) because turn-on spikes occur before the switch closes the signal circuit, and (2) because the output impedances of sources such as feedback amplifiers, while low for signal frequencies, may not be low for the fast spike waveforms.

Pedestals can be substantially eliminated by balanced control voltages (Fig. 5-3) or d-c balancing. Switching spikes cannot be reduced by much more than a factor of 2 or 3 through injection of inverted control voltages unless only d-c is switched, because the switch-junction capacitances change with the signal voltage. Switching spikes will be short for small coupling capacitances but can be as large as the full control-step amplitudes with resistive loads (Fig. 5-3). This is especially serious in chopper amplifiers (Sec. 3-36), where spikes can overload the amplifier. Such objectionable switching spikes may have to be reduced (at some expense in switching speed) through the use of slower control waveforms or capacitive loading in the signal circuit.

(f) A number of practical circuits employing feedback amplifiers to suppress effects of forward resistance and/or d-c offset in electronic switches will be described in Sec. 5-5.

5-2. Diode Switches.[1,9-11] The most generally useful diode switches are various types of *diode-bridge switches* (Fig. 5-4). Four- and six-diode bridges have also been used as shunt switches (clamping circuits) and in series-shunt combinations. Diode switches are best realized with modern fast-recovery low-leakage silicon-junction diodes in naturally matched monolithic sets.

To turn any diode bridge fully OFF, one requires control voltages as large or larger than the voltage to be switched. *To turn any diode-bridge switch* ON, *the control circuit must supply a current at least as large as the signal current required to flow through the switch,* or the bridge will act as



(*a*) (*b*)

(*c*) (*d*)

Fig. 5-4*a* to *d*. Diode-bridge switches. r_L can be an operational-amplifier input resistor. Circuits *b*, *c*, and *d* combine shunt and series switching and turn off quickly. In *a*, *b*, and *c*, half of each circuit can be used to switch positive input voltages (solid lines) or negative input voltages (dashed lines); but the leakage and spike cancellation inherent in the symmetrical circuits is desirable even with input voltages of one sign. Circuit *d* does not permit half-bridge operation and requires careful balancing of E/r and $-E'/r$ to minimize offset in the ON condition.

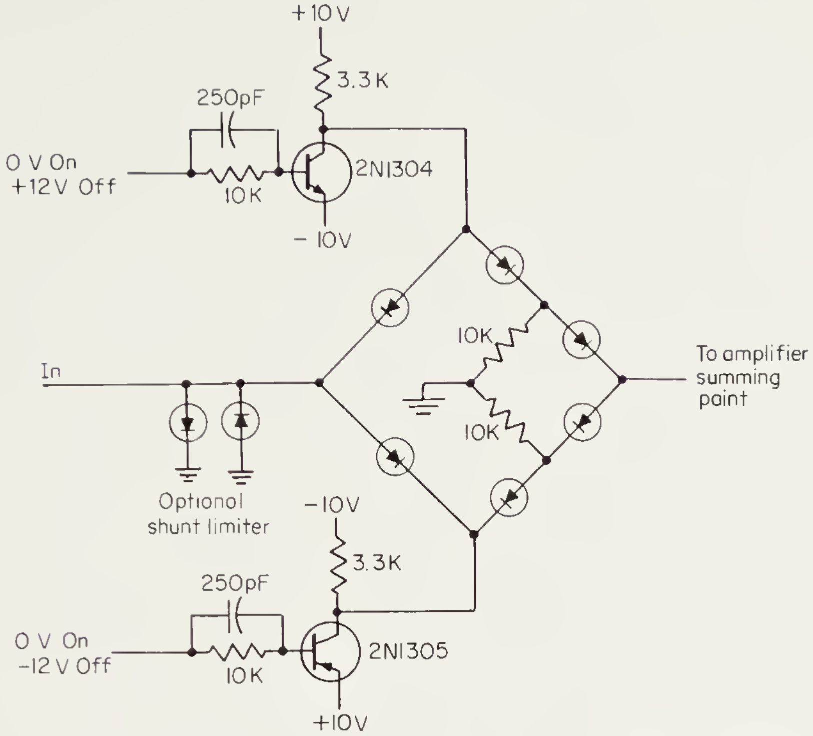


Fig. 5-4*e*. A combined diode-bridge/transistor switch for track-hold or integrator switching in ±10-volt computers. Two-diode networks in the output legs of the bridge reduce leakage and capacitance. (*University of Arizona.*)

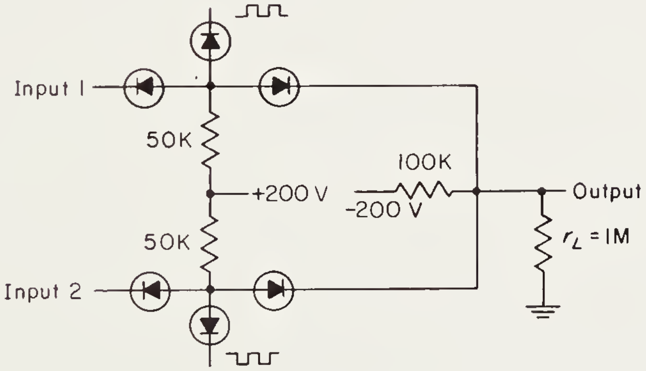


Fig. 5-4*f*. SPDT switch developed from the selector circuits of Fig. 4-5.

a limiter. This is an important consideration if the switch is to supply capacitor charge or discharge current (Sec. 5-17).

The *six-diode bridge switches* of Fig. 5-4*b* and *d* combine shunt and series switching to make both resistive and capacitive leakage from the signal source essentially negligible (Sec. 4-5). Switch forward resistance is about 50 ohms for silicon-junction diodes. Silicon-junction output-diode leakage currents can be below 0.05 μA at room temperature. The symmetrical bridge circuits further reduce the effective leakage current by a factor of at least 3, especially if the bridge diodes are selected or fabricated to match over the range of operating temperatures. The remaining errors are caused by switching spikes due to uncanceled capacitive leakage from the control circuit and differences in the switching times of the two bridge-circuit branches and are especially serious when the switch supplies current to a capacitor (Sec. 5-17). The switching spikes are not readily reduced by trimmer capacitors shunting the bridge diodes; one can try to use trimmer capacitors to vary the control-step rise and fall times on one side of the bridge.

For a six-diode switch operated with a low-impedance control circuit, an equivalent circuit similar to that of Fig. 4-8c[1,9] indicates that the turn-off time constant is of the order of $C_B R_F$; so that six-diode switches can turn off quickly. The turn-on time will depend on the resistance in the signal circuit; turn-on times within 0.5 μsec are readily obtainable.

The reliable and inexpensive diode-bridge switches can switch voltages exceeding 100 volts.

In designing diode switches, note:

1. When a four- or six-diode switch is OFF, *all four* (not just two) series diodes must be OFF. Therefore, the positive and negative control voltages holding the switch OFF in Fig. 5-4 must be *larger than the largest positive and negative signal-voltage excursions.*
2. To keep the diode switches of Fig. 5-4*a*, *b*, and *d* ON without signal limiting (Sec. 4-6), the current supplied by the control voltages in Fig. 5-4*a* and by E and $-E$ in Fig. 5-4*b* and *c* must be *at least as large as the largest signal current supplied to the load.* This current-limiting effect must be observed especially with capacitive loads.

5-3. Bipolar-transistor Switches.[1–6,12–22] **(a) Static Characteristics.** The collector characteristics of junction transistors (Fig. 5-5) suggest the use of both *npn* and *pnp* transistors as *bidirectional* electronic switches permitting control of the emitter-collector current i_C by the—usually smaller—base current i_B. Transistor switches can, therefore, work with smaller control currents than diode switches, and better isolation of control and signal circuits becomes possible. Referring to Fig. 5-5*b*, a transistor switch is ON when a sufficiently positive base current i_B causes operation in the *saturation region;* for zero or negative i_B, the transistor is *cut off,* and the switch is OFF. Since transistors, unlike vacuum tubes, are controlled by current rather than voltage, *each transistor-switch design must provide a control-current return path which will not interfere with the signal circuit.*

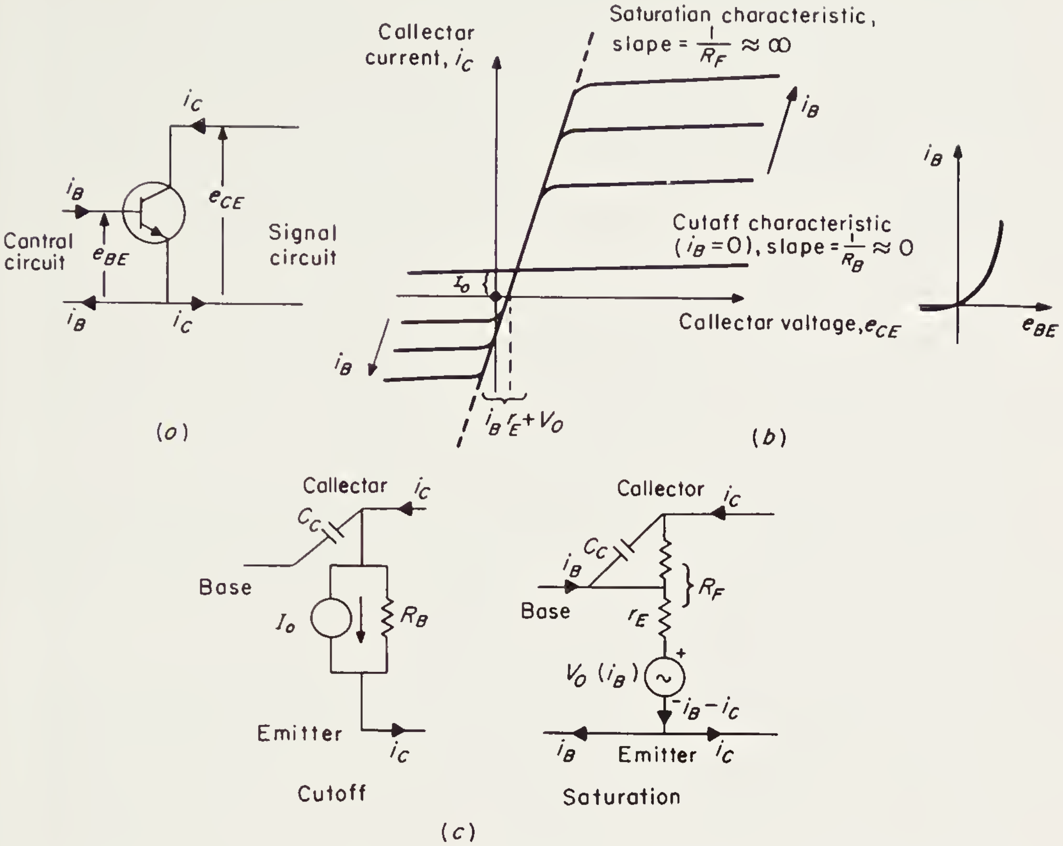


Fig. 5-5. *npn* transistor switch (*a*), transistor characteristics (*b*), and simplified equivalent circuits (*c*).

Actual transistor collector characteristics show that:

1. The reciprocal slopes $\Delta e_{CE}/\Delta i_C = R$ of the saturation and cutoff characteristics respectively yield *forward resistances* $R = R_F$ between 10 and 100 ohms and *back resistances* $R = R_B$ between 1 and 500M, just as in the case of junction-diode switches.
2. The saturation and cutoff characteristics in Fig. 5-5*b* do not pass exactly through the coordinate origin, as ideal switch characteristics would. The resulting offset errors are interpreted as follows: the saturation characteristic is displaced horizontally by a positive *saturation offset voltage* $i_B r_E + V_0(i_B)$ of 10 to 500 mV; and the cutoff characteristic is displaced vertically by a positive *leakage current* I_0 between 0.02 and 50 nA at 25 deg C.

This situation is summarized by the simplified equivalent circuits of Fig. 5-5*c*. The offset sources are often disregarded in the design of digital switches, but they are important for analog switching; I_0, in particular, varies exponentially with temperature, and it will be necessary to specify low-leakage transistor types.

(b) The Inverted Transistor as a Switch. If an *npn* or *pnp* junction transistor is *inverted*, i.e., if one interchanges the roles of emitter and collector as in Fig. 5-6*a*, the device still operates as a transistor whose characteristics are shaped like those in Fig. 5-5*b*, although they are *compressed vertically.* More specifically, experimental and theoretical investigation of the normal and inverted transistor characteristics reveals that:

1. The current amplification factor $\beta = \partial i_C/\partial i_B$ is decreased by a factor of 5 to 100 in the inverted connection, and $-\log_e \alpha = -\log_e (\partial i_C/\partial i_E)$ is decreased by about a factor of 10. The inverted connection is, therefore, not suitable for amplification or for power switching with small control currents.
2. The switch forward resistance (reciprocal slope of the saturation characteristic) is only slightly higher in the inverted connection.
3. The inverted transistor connection typically decreases the offset voltage $V_0 + i_B r_E$ by a factor of at least 10.

NOTE: In earlier (germanium, silicon-alloy) switching transistors, the leakage-current offset in the OFF state (as well as the offset voltage in the ON state) was substantially reduced by the inverted connection.[1-3,12-15] This is *not* true for modern silicon switching transistors, where what little leakage is left is due mainly to surface leakage and charge generation rather than to diffusion in the junction.

Transistors suitable for analog switching are silicon epitaxial-planar types combining low saturation resistance and leakage with relatively small junctions for high switching speed. For typical silicon transistors of this kind (e.g., Motorola 2N2220, 2N2369, MPS6512) used in the inverted connection at 25 deg C, we find:

Inverted-saturation voltage $|i_B r_E + V_0(i_B)| < 1$ mV ($i_B = 0.5$ mA)
Inverted leakage current $|I_0| < 0.1$ nA ($i_B \approx 0$)
Inverted-saturation resistance ("dynamic resistance") $R_F < 25$ ohms ($i_B = 0.5$ mA)
Back resistance $R_B > 1$ to 100M

Since the emitter-base and collector-base junctions of a junction transistor behave essentially like junction diodes (Sec. 4-2), a cutoff transistor switch exhibits nonlinear emitter-base and collector-base capacitances (typically 5 to 15 pF for 1 volt reverse bias) and turn-on storage time (below 0.1 μsec for switching transistors). Transistor capacitances and storage time are, approximately, inversely proportional to the quoted alpha-cutoff frequency f_α, since all these quantities depend on the junction dimensions. f_α is, therefore, a useful measure of the attainable switching speed. Since the collector-base junction is larger

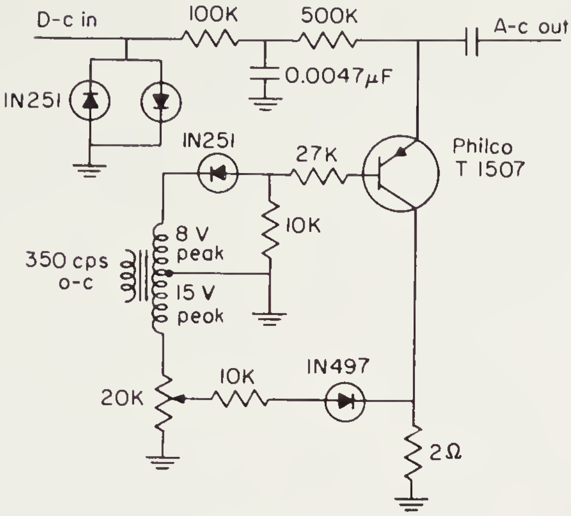
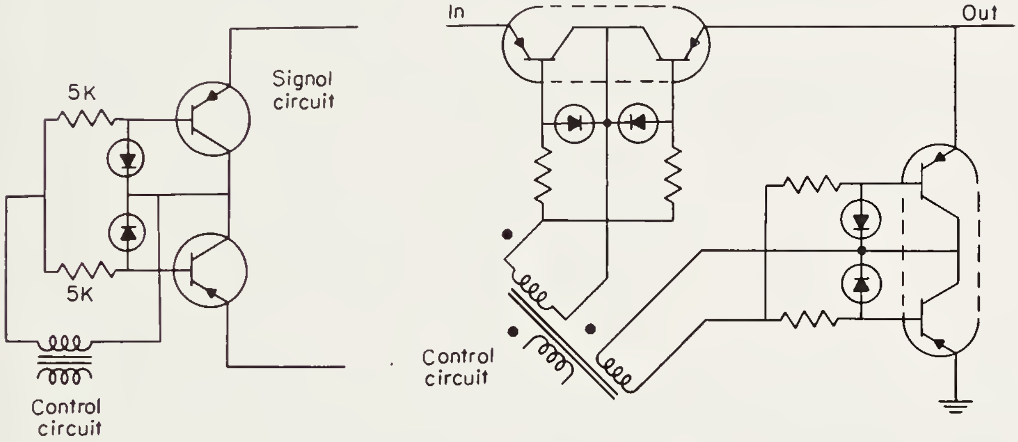


Fig. 5-6*a*. An accurate inverted-transistor chopper. The diode-limited input voltage cannot exceed ±0.7 volt. The OFF base voltage is approximately zero. Offset is less than 3 μV/deg C, or 50 μV/100 hr at constant temperature.[18]



(*b*) (*c*)

Fig. 5-6*b*, *c*. Transistor switches employing matched inverted-transistor pairs with floating control circuits.

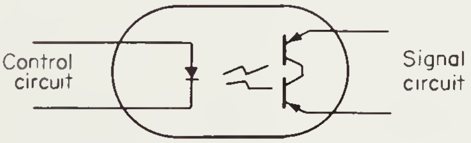


Fig. 5-6*d*. All control-signal interference is eliminated in this light-actuated dual-transistor switch. Unfortunately, response lags in the light source (infrared-emitting diode) prevent switching times shorter than a few microseconds.

than the emitter-base junction, an inverted transistor has a higher input capacitance than a normal transistor, but typical switching times are still well below 1 μsec for switching transistors with $f_\alpha > 100$ MHz, and 25 nsec can be attained. *Speedup capacitors* across base-current-setting resistors will improve switching speeds by supplying fast control-current spikes when the control voltage is reversed.

More specifically, the offset voltage $V_0(i_B)$ is approximately proportional to $1/i_B$; so that plots of the total inverted-saturation offset $i_B r_E + V_0(i_B)$ vs. i_B, which are supplied for most switching transistors, show a shallow minimum. In the vicinity of this minimum, $i_B r_E + V_0(i_B)$ is fairly insensitive to changes in i_B, and the offset temperature coefficient is, fortunately, close to zero (typically positive and below 2 to 10 μV/deg C for practical base currents).[4,14] As a result, it becomes practical to balance the saturation offset partially by a d-c voltage, or by the saturation offset of a second transistor (Fig. 5-6*b* to *d*).

(c) Practical Switching Circuits (see also Secs. 5-16, 5-17, and 5-18). *To turn a transistor switch* OFF, *the base-control circuit must supply a voltage at least large enough to reverse-bias both the emitter-base junction and the collector-base junction. To turn the transistor switch* ON, *the base current must exceed the signal current divided by* β_N (*normal connection*) *or* β_I (*inverted connection*).

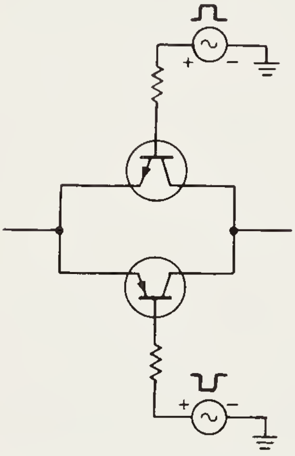


Fig. 5-7. A complementary-transistor combination providing a convenient base-current return path.

Figure 5-6*a* shows an inverted *pnp* transistor operating as a low-level shunt switch or *transistor chopper* useful for low-level modulation at frequencies below 1 KHz (as in d-c amplifier stabilization, Sec. 3-36). A diode limiter keeps the input signal within about ±0.6 volt. Note that the transistor is cut off with zero base current to obtain a leakage current I_0 below 0.1 nA. An adjustable d-c voltage intended to balance the saturation offset is applied in the ON condition only, so that it cannot cause leakage variations in the nonconducting state of the switch. Errors due to capacitive leakage are minimized through operation at chopper frequencies below 500 Hz, although the circuit works at much higher frequencies. Switching errors obtained with this switch amount to less than 100 μV d-c referred to the input over a temperature range between 10 and 50 deg C. The effects and partial neutralization of *switching spikes* in transistor choppers are discussed in Sec. 5-1*e*.

The circuits of Fig. 5-6*b* and *c* combine *matched pairs of inverted transistor switches* for partial offset cancellation. Two matched switching transistors can have a common collector-base structure to improve thermal matching, to reduce the number of thermoelectric junctions, and to minimize interference between control and signal currents.[16] Unfortunately, such dual-transistor circuits require either floating-transformer control circuits or relatively complicated floating control multivibrators[17] to provide a base-current return path. The *npn-pnp* combination of Fig. 5-7 provides a convenient base-return path at the expense of an additional transistor.

Figure 5-8*a* and *b* shows bipolar-transistor SPDT switches used, for instance, for polarity changing and in digital-to-analog conversion

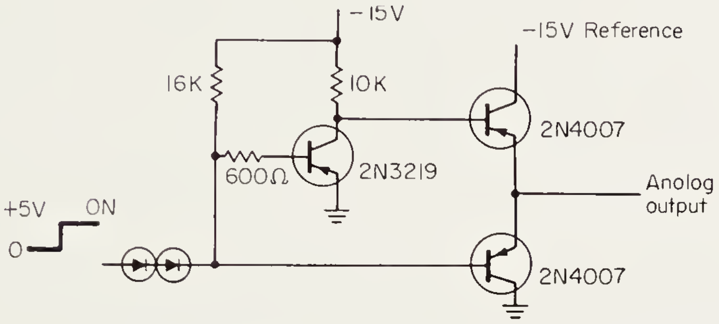


Fig. 5-8*a*. SPDT switch used to switch an analog output voltage between a −15-volt reference and ground (DAC bit switch, see also Sec. 5-7). (*Crystalonics, Inc., Cambridge, Mass.*)

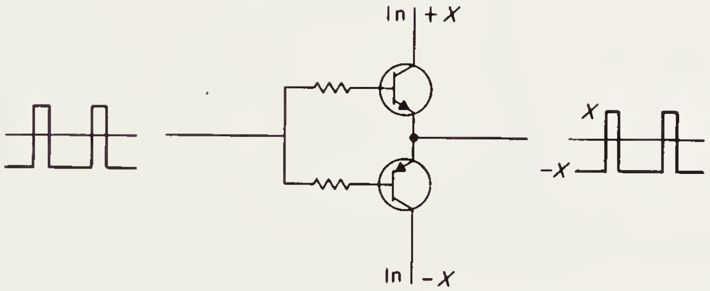


Fig. 5-8*b*. An SPDT switch employing complementary inverted transistors.

(Sec. 5-7). Both circuits combine shunt and series switching. Unless the base control currents are supplied from floating control circuits, the base currents of the two transistors will vary with the signal voltages, and the saturation-offset voltages will also change and cannot be accurately canceled. Storage times in the two transistors in Fig. 5-8*b* will also change with the currents produced by the two input voltages X_1 and X_2, so that it becomes practically impossible to switch both transistors simultaneously. This results not only in switching spikes but in actual shorting of the signal sources. For this reason, and to minimize the effect of base-current leakage into the signal circuit, such switches must have low-impedance signal sources.

The control-current-return problem in Fig. 5-8*b* could be eliminated by a floating-transformer control circuit in the manner of Fig. 5-6.

Figure 5-9 shows a switching circuit designed to permit arbitrarily long OFF and ON periods with all the advantages of floating-transformer control. This circuit employs a transformer-rectifier source excited by an RF oscillator switched on and off by the control-logic signal.[19]

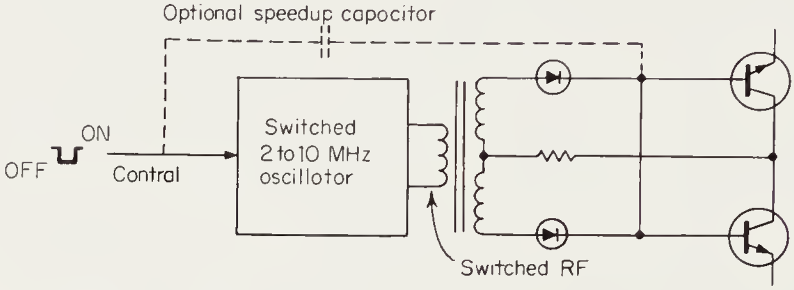


Fig. 5-9. A matched-transistor-pair switch with a floating transformer-rectifier control-current source whose RF input is turned on and off to control the switch. Only surges from the optional speedup capacitor can pass through signal source or load. Switching times of 0.1 μsec are possible with the speedup capacitor.

5-4. Field-effect-transistor Switches. (a) Junction Field-effect Transistors (JFETs). The *n*-channel junction field transistor of Fig. 5-10*a* acts, for *zero* gate-to-source voltage e_{GS}, like a passive silicon resistor. In this ON state, the FET passes current in either direction, and *there is no voltage offset.* Increasingly negative e_{GS} will "pinch" the flow of negative carriers in the channel, until the current ceases at a *pinch-off voltage* $e_{GS} = e_1$ (−8 volts in Fig. 5-10*b*). In this OFF condition, one has small leakage currents (typically less than 0.01 to 0.3 nA at 25 deg C, doubling every 10 deg C) across the gate-source and gate-drain junctions; these junctions also act as nonlinear capacitances. Figure 5-10*c* shows simple equivalent circuits in the ON and OFF states. JFET forward resistances range between 3 and 300 ohms.

(b) Metalized-oxide Semiconductor Field-effect Transistors (MOSFETs). The insulated-gate field-effect transistor illustrated in Fig. 5-11 is OFF for zero gate-to-source voltage e_{GS} and reduces its forward resistance to turn ON as e_{GS} is made more negative (*enhancement mode*). Equivalent circuits similar to Fig. 5-10*c* apply, but with the gate electrode insulated by a thin layer of silicon oxide, leakage is essentially negligible, and the gate-source and gate-drain capacitances are small (1 pF). On the other hand, forward resistances R_F are rarely smaller than 150 to 500 ohms. R_F varies not only with current and temperature

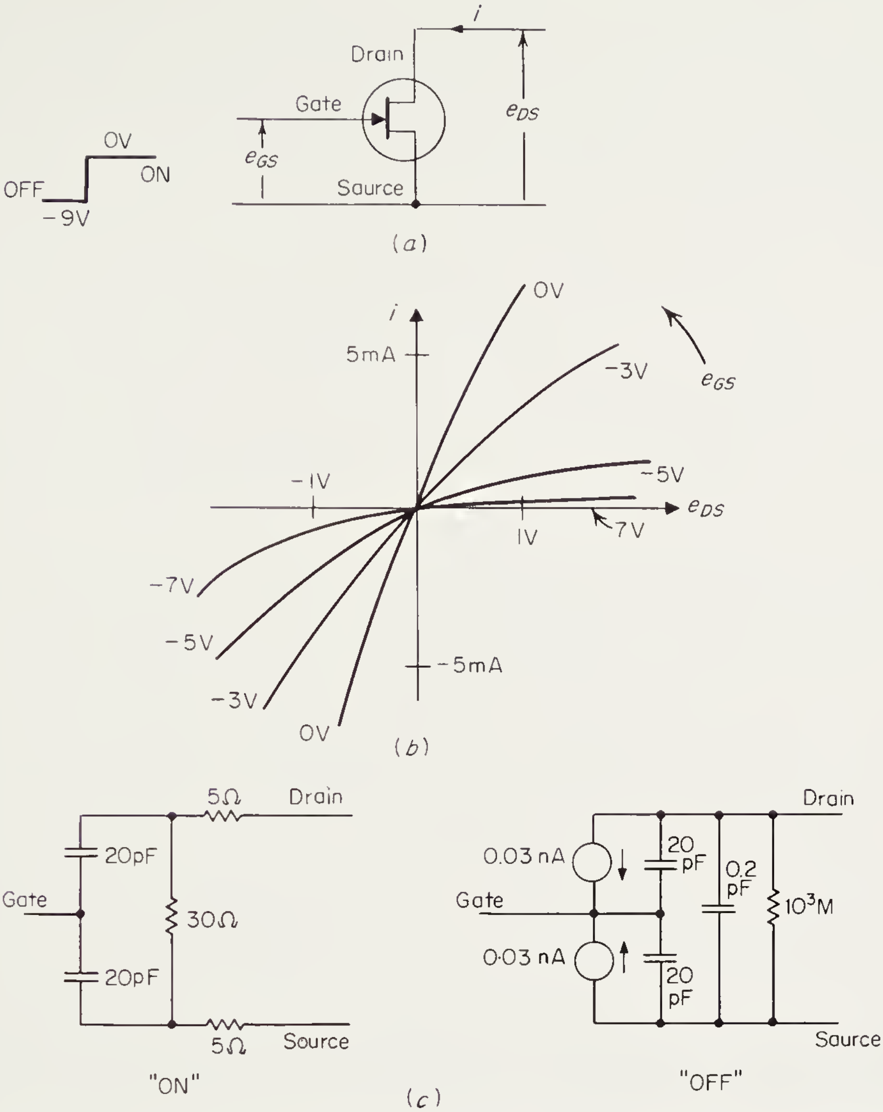


Fig. 5-10. *n*-channel JFET (*a*), characteristic curves (*b*), and simple equivalent circuits (*c*). Voltages and junctions would be reversed for a *p*-channel JFET.

but also with the control voltage on the gate, and thus possibly with power-supply voltages.

Many practical switching MOSFETs have an enhancement-mode ON condition and a *depletion-mode* OFF condition. A *p*-channel device of this type might be ON with $e_{GS} = -6$ volts and OFF with $e_{GS} = +8$ volts.

NOTE: In designing MOSFET switches, the *substrate-to-channel junction must be back-biased for all signal-voltage excursions.* For the *p*-channel MOSFET of Fig. 5-11, this means that the substrate must be kept more positive than any signal peak. For inputs below 0.6 volt, one can simply connect substrate and source.

(c) Practical Switching Circuits. Figure 5-12 shows a practical JFET switch. The coupling diode normally prevents forward biasing of the gate (-to-channel) junction. To keep the switch OFF, the driver-amplifier output must be 8 to 12 volts more negative than the largest signal excursion at *S*. To turn the switch ON, the driver output is made at least 0.6 volt more positive than any signal-voltage excursion. The coupling diode then cuts off, and resistor *R* ensures zero gate-to-source voltage.

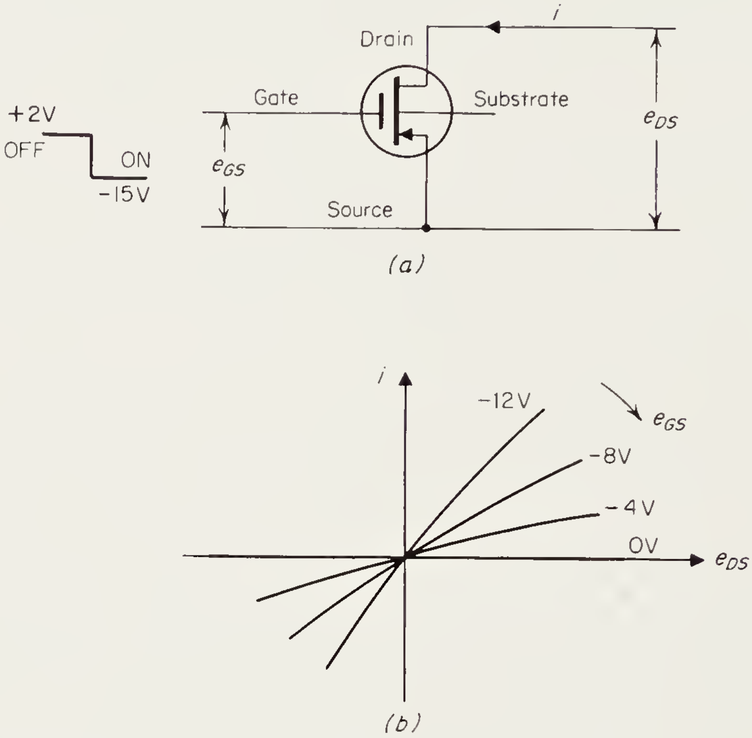


Fig. 5-11. *p*-channel enhancement-type MOSFET (*a*) and characteristic curves (*b*). The substrate, made of *n*-type material, forms a junction with the *p* channel and must not be negative with respect to the channel to keep this junction cut off. Voltages and junctions would be reversed for an *n*-channel enhancement-type MOSFET.

Even if the gate-to-source resistor *R* is omitted, the positive turn-off spike transmitted through the diode capacitance can momentarily forward-bias the gate junction to approximate zero gate-to-source voltage for some time. Such circuits are useful for fast periodic switching. To optimize the turn-on charge, one can parallel the diode with a small capacitor (Fig. 5-13), or the diode may be replaced by the gate junction of a second JFET (with source and drain connected together).

Figure 5-13 shows a complementary-FET SPDT switch. Figure 5-14 illustrates the recommended method for switching operational-amplifier inputs with FETs (see also Fig. 5-15*b*). The combination of a series FET and an inexpensive silicon shunt transistor or of series and shunt FETs yields several significant advantages:

1. *Combined series-shunt switching* (Sec. 5-1) minimizes signal leakage through the FET junction capacitances.

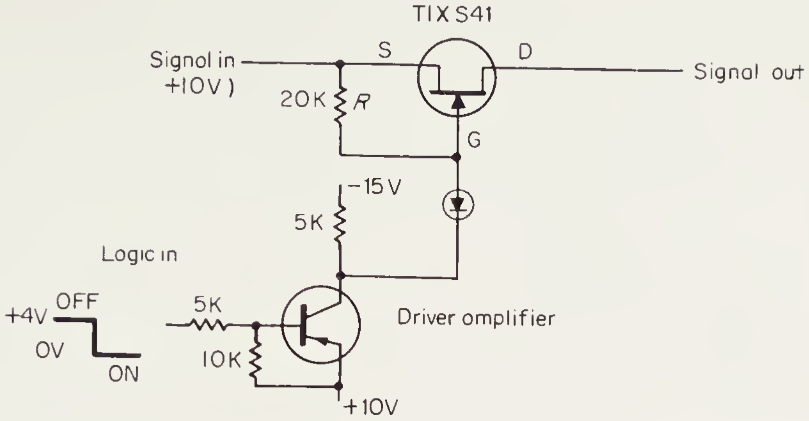


Fig. 5-12. JFET switch with gate-to-source resistor. The value of R is chosen experimentally to produce the desired turn-on time.

2. The voltage excursions of the FET source and drain are very small both when each FET is ON and when each FET is OFF with its source shorted to ground potential by the second switch. *This simplifies the FET gate-drive requirements.*
3. The impedances seen by the source in the ON and OFF states are roughly similar in Fig. 5-14*a* and *b* and can be accurately matched in Fig. 5-14*c*. This is important in the ladder circuit of Fig. 5-21.

The shunt-transistor leakage is practically negligible, and the shunt-transistor voltage offset is kept from the summing point by the open FET. TTL-gate outputs have been used directly for shunt switching.

(d) Comparison of JFET, MOSFET, and Bipolar-transistor Switches. FET switches do not require current from their control-signal sources, but they usually need control-voltage swings larger than those supplied by most digital logic, so that special driver amplifiers are required.

Commercially available monolithic integrated circuits combine several MOSFET switches with their MOSFET driver amplifiers.

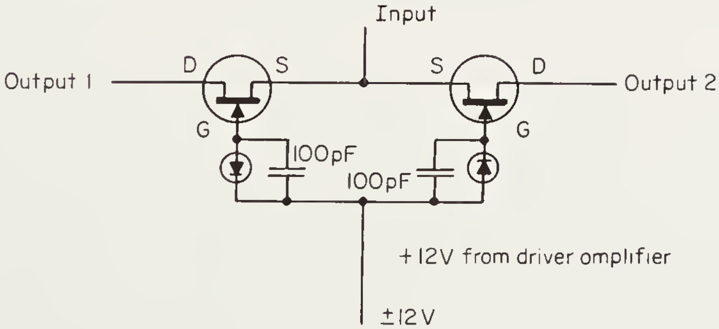


Fig. 5-13. Complementary-JFET SPDT switch without gate-to-source resistors. This circuit is intended for periodic switching; the coupling capacitors are chosen experimentally.

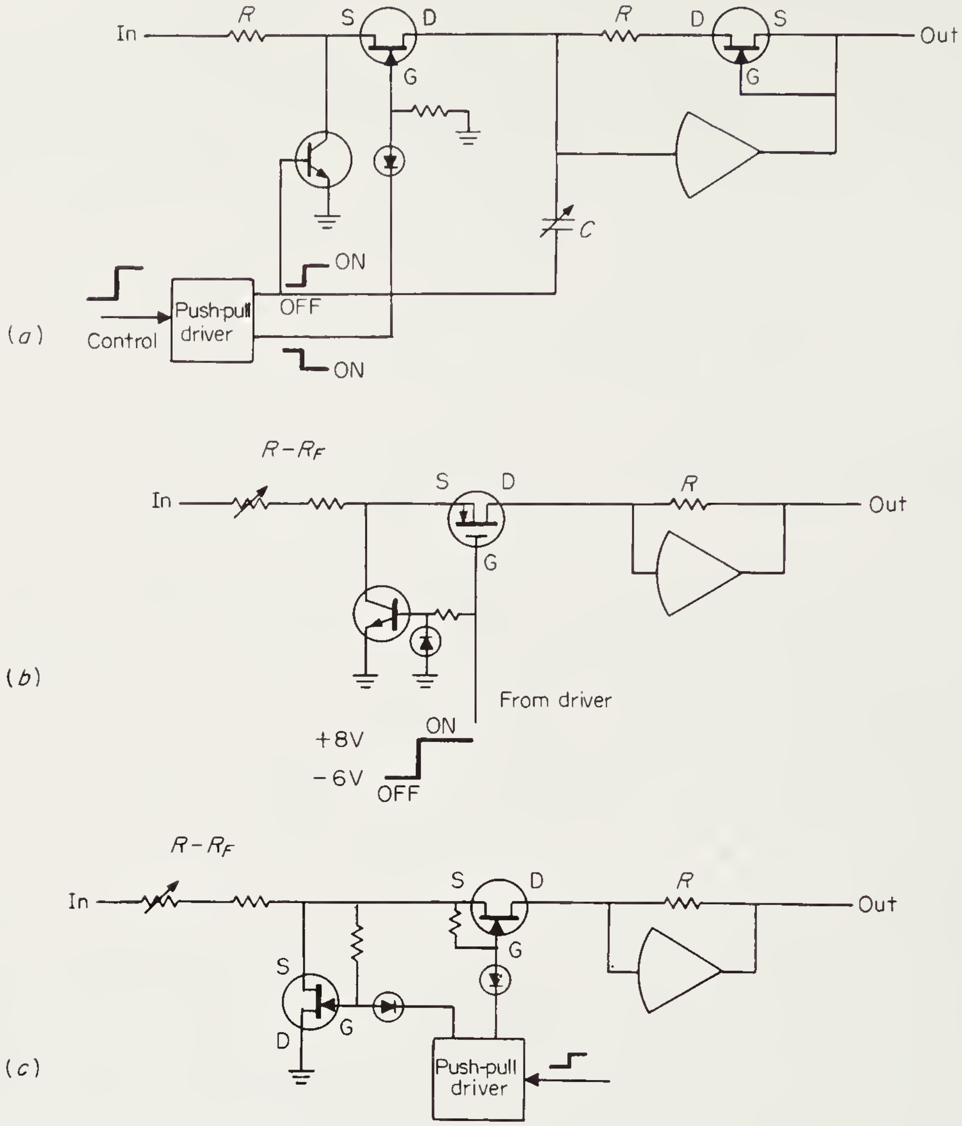


Fig. 5-14. Series-shunt switching of operational-amplifier input current. The temperature variation of the JFET resistance in (*a*) is compensated by a similar JFET in the feedback circuit. *C* is a trimmer for partial switching-spike cancellation. Circuit (*b*) employs an enhancement/depletion MOSFET. Circuit (*c*) permits accurate matching of the impedance seen by the source in the ON and OFF states.

Compared to bipolar transistors, FET switches are practically offset-free but have, at least in inexpensive units, higher forward resistances. Resistances of *n*-channel JFETs are slightly lower than those of *p*-channel JFETs of similar geometry. Unfortunately, JFET junction capacitances increase with decreasing R_F (larger junctions). High-priced units may have $R_F = 4$ ohms at 25 deg C with 20-pF junction capacitances. $R_F = 30$ ohms is more typical. As prices of high-quality FETs and integrated-circuit drivers decrease, they can be expected to replace bipolar transistors in most switching applications.[80]

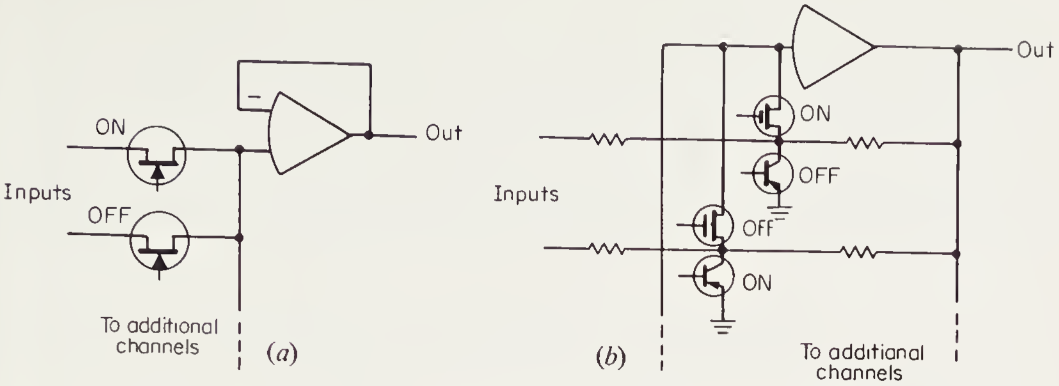


Fig. 5-15. Two multiplexer circuits which minimize errors due to switch resistances. The amplifier in (*a*) is connected as a high-impedance follower (Sec. 3-30).

FET-switch forward resistances *increase with temperature* at rates between 0.1 and 0.7 percent/deg C, depending on type and device geometry (refer to manufacturers' specifications). R_F also changes slightly with current (of the order of 2 percent/mA, Figs. 5-10*b* and 5-11*b*).

MOSFETs have higher forward resistances (100 to 800 ohms) but smaller capacitances and even less leakage than JFETs. The ON gate voltage of MOSFETs in the enhancement mode affects their forward resistance, and the switch designer should check on the magnitude of this effect.

Unlike enhancement-type MOSFETs, JFET switches are ON when control-circuit power is OFF; this may or may not cause trouble. Simple MOSFETs (without diode gate protection) are easily damaged by electrostatic discharges.

Finally, switching times between 0.2 and 2 μsec can be obtained with either JFETs or MOSFETs.

5-5. Some Improved Switching Circuits. If it becomes desirable to suppress the (usually nonlinear) forward resistance of a given electronic

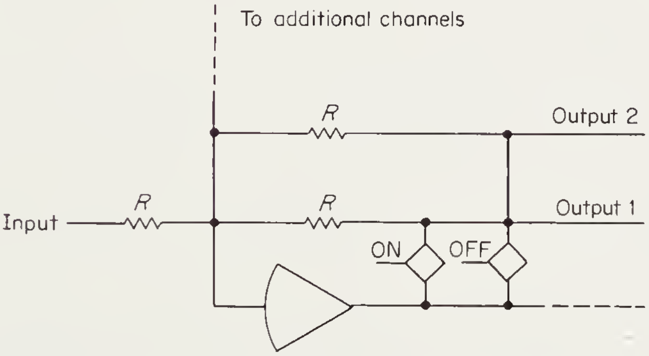


Fig. 5-16. "Tail-switching" demultiplexer divides offset errors as well as switch forward resistance by the amplifier loop gain, but switch terminals move through the full signal-excursion range.

switch, it may be possible to:

1. Place the switch in a high-resistance circuit (Fig. 5-15*a*)
2. Drive the switch with a current source (Sec. 5-8)
3. Use the switch inside a feedback loop (Figs. 5-15*b* and 5-16)

See Sec. 5-17 for applications of switched amplifiers (nonsaturating analog switches) for fast high-current switching.

DIGITAL-TO-ANALOG CONVERTERS

5-6. Basic Conversion Circuits. Representation and Conversion of Negative Numbers (see also Table 5-2). Figure 5-17 shows simple *digital-to-analog converters* (*DACs*) capable of multiplying an analog input voltage X_1 by an n-bit nonnegative binary fraction

$$Y_D = \frac{1}{2}a_1 + \frac{1}{4}a_2 + \cdots + \frac{1}{2^n}a_n \tag{5-2}$$

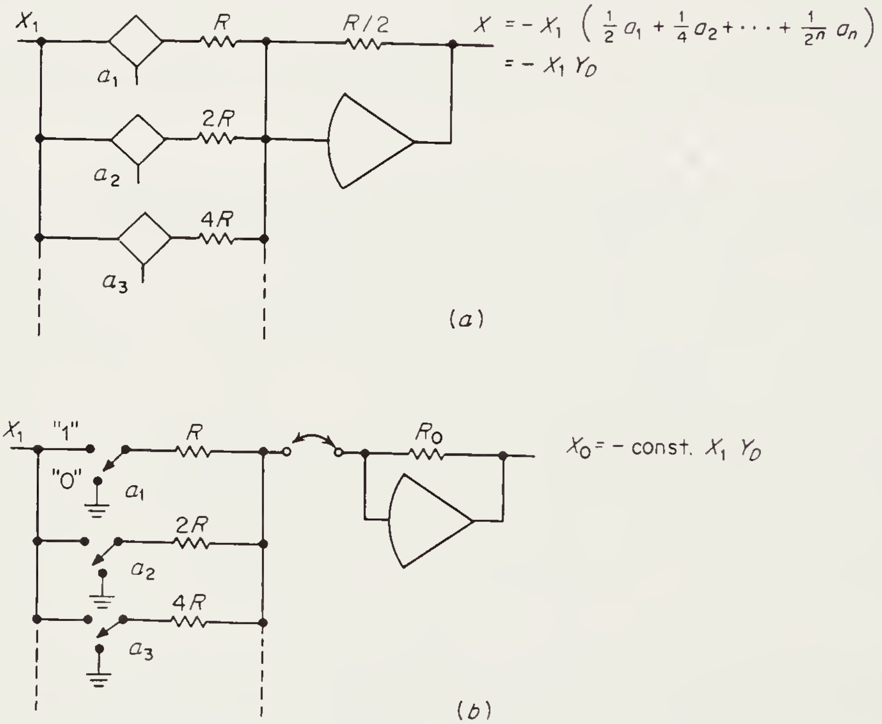
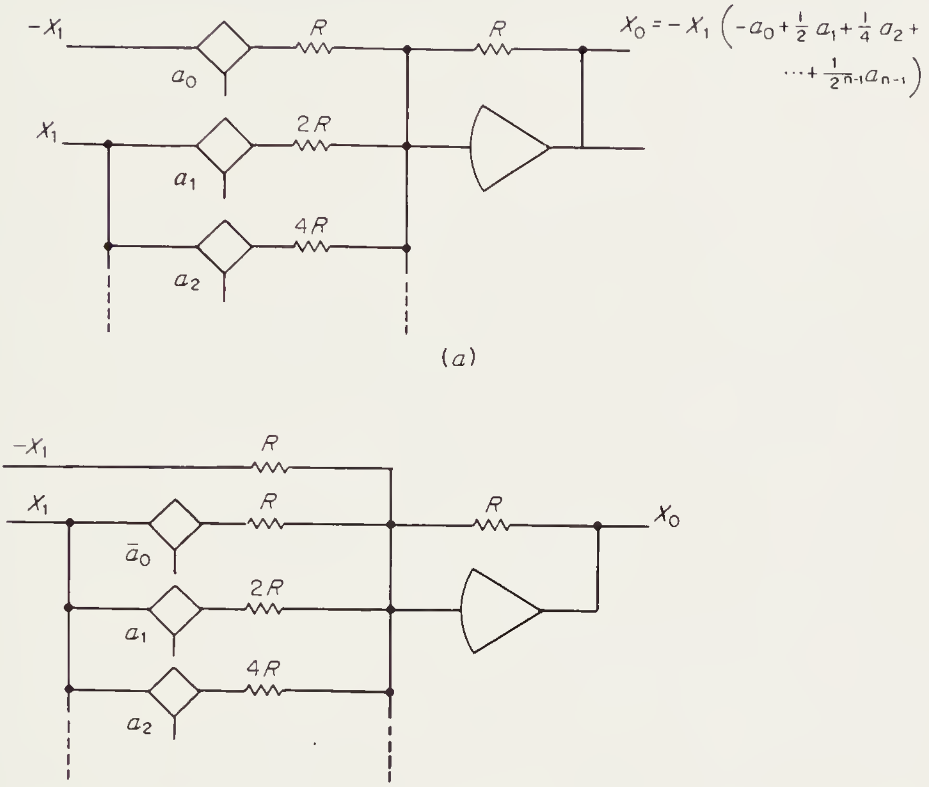

X_1
R
$R/2$
$X = -X_1\left(\frac{1}{2}a_1 + \frac{1}{4}a_2 + \cdots + \frac{1}{2^n}a_n\right)$
$= -X_1 Y_D$
a_1
$2R$
a_2
$4R$
a_3
(a)
X_1
"1"
R
R_0
$X_0 = -$ const. $X_1 Y_D$
"0"
a_1
$2R$
a_2
$4R$
a_3
(b)

Fig. 5-17. Simple digital-to-analog converters (DACs) for n-bit nonnegative binary numbers

$$Y_D = \frac{1}{2}a_1 + \frac{1}{4}a_2 + \cdots + \frac{1}{2^n}a_n$$

where $a_k = 0$ or 1 ($k = 1, 2, \ldots, n$). The kth switch is ON if the corresponding binary digit a_k is 1. Circuit (*b*) has less feedthrough when switches are OFF; this circuit also has *constant source impedance*, so that it could drive a simple load resistance r_L instead of an operational-amplifier summing point.

The kth switch is ON if the corresponding binary digit a_k is 1, so that $0 \leq Y_D < 1$. a_1 is the *most significant bit* (MSB) of Y_D, and a_n is the *least significant bit* (LSB). The analog input X_1 may be a fixed reference voltage (*fixed-reference DAC*), or a variable (*multiplying DAC, MDAC*). MDACs are more difficult to design and more expensive, because switch-resistance effects, and possibly switch offset voltages, will vary with X_1.


$-X_1$
R
R
$X_0 = -X_1\left(-a_0 + \frac{1}{2}a_1 + \frac{1}{4}a_2 + \cdots + \frac{1}{2^{n-1}}a_{n-1}\right)$
a_0
X_1
$2R$
a_1
$4R$
a_2
$-X_1$
R
X_1
R
R
X_0
$\bar{a}_0$
$2R$
a_1
$4R$
a_2

(a)

(b)

Fig. 5-18. Conversion of two's-complement-coded positive or negative binary fractions

$$Y_D \equiv -a_0 + \frac{1}{2}a_1 + \frac{1}{4}a_2 + \cdots + \frac{1}{2^{n-1}}a_{n-1}$$

If X_1 is a fixed reference voltage, the scheme of Fig. 5-18*b* requires only switches passing current in one direction. Either scheme also works with ladder networks.

Figure 5-18 illustrates the conversion of a *positive or negative two's-complement-coded n-bit binary fraction*

$$Y_D \equiv -a_0 + \frac{1}{2}a_1 + \frac{1}{4}a_2 + \cdots + \frac{1}{2^{n-1}}a_{n-1} \tag{5-3}$$

The most significant bit (MSB) is now the *sign bit* a_0, and the least significant bit (LSB) is a_{n-1}. Note that $-1 \geq Y_D < 1$. Starting at

-1, Y_D would increase as follows:

$$\left.\begin{array}{l}
100\cdots00 = -1\\
100\cdots01 = -(1-2^{1-n})\\
100\cdots10\\
\cdots\cdots\cdots\\
111\cdots11 = -2^{1-n}\\
\hline
000\cdots00 = 0\\
\hline
000\cdots01 = 2^{1-n}\\
000\cdots10\\
\cdots\cdots\cdots\\
011\cdots11 = 1-2^{1-n}
\end{array}\right\}\quad \text{(TWO'S-COMPLEMENT CODE)} \qquad (5\text{-}4)$$

Four-quadrant multiplication of Y_D and an analog input X_1 requires both X_1 and $-X_1$ as input voltages.

If more than 5 or 6 bits are needed, the resistances 2^kR in Figs. 5-17 and 5-18 may become uncomfortably large (see also Sec. 5-9). The network impedance levels can be reduced nicely if resistances are replaced with *T-network transfer impedances* in the manner of Fig. 5-19. Although this requires additional resistors, most of them can have the same value. In the operational-amplifier circuits of Fig. 5-19, the DAC network need not constitute an accurately constant source impedance, so that SPST switches can be used to ground each T junction with minimum feedthrough.

While the requirement of a given static accuracy at low cost favors *high* DAC-network resistances to make switch resistances less critical, each DAC must settle to within the specified accuracy after switching, and the required settling time will be proportional to some circuit resistance (Fig. 5-27). The sum of this settling time and the (usually smaller) switching time constitutes the *conversion time*, whose reciprocal is the maximum *conversion rate* still permitting the specified accuracy. High conversion rates, and also low MDAC phase shift, favor *low* DAC-network resistances. Hence DAC design will require a compromise between accuracy, conversion rate and/or phase shift, and switch costs (see also Sec. 5-9).

Digital-to-analog converters employing switched-capacitance networks to decode digital data arriving in serial form (i.e., bit by bit) are described in Refs. 4, 5, and 65. Figure 5-28 shows a simple example.

5-7. Voltage-ladder Digital-to-analog Converters. The basic *binary voltage-ladder network* illustrated in Fig. 5-20 is equivalent to a voltage source of internal impedance R and source voltage

$$\frac{1}{2}\left(a_1 + \frac{1}{2}a_2 + \cdots + \frac{1}{2^n}a_n\right)$$

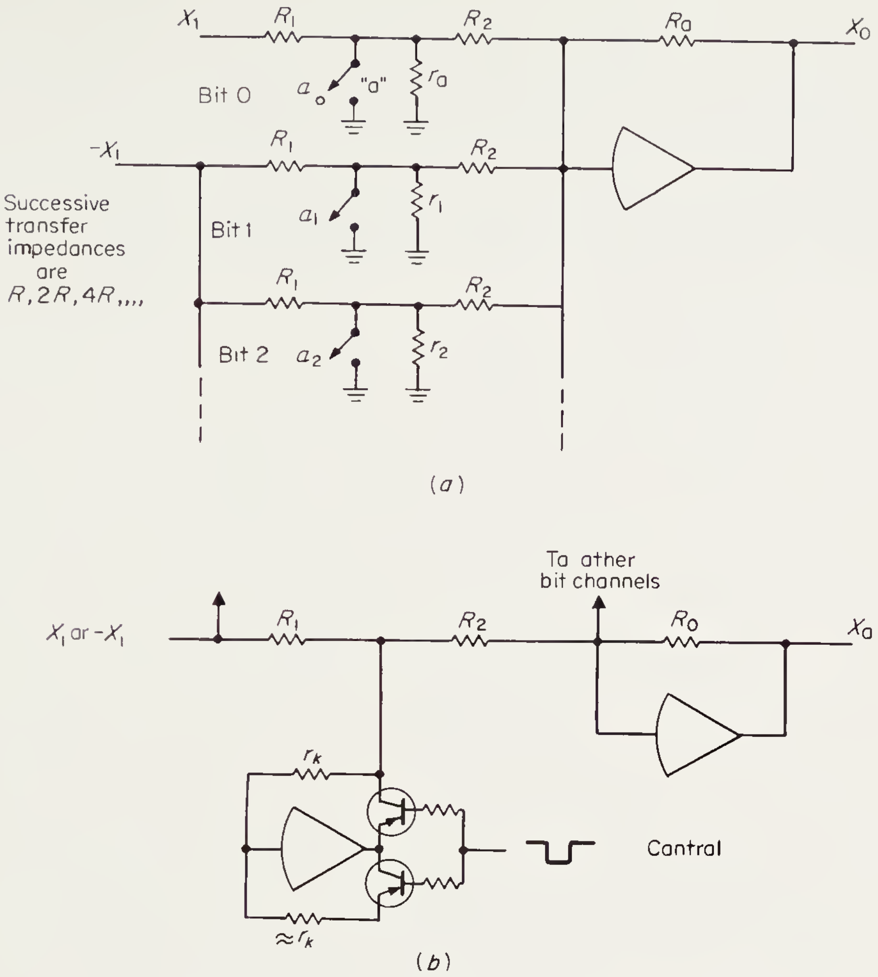


Fig. 5-19. This summing-network MDAC employs T-network transfer impedances with

$$Z_k = R_1 + R_2 + \frac{R_1R_2}{r_k} = 2^{-k}R \qquad (k = 0, 1, 2, \ldots, n-1)$$

instead of simple resistances to permit low circuit-impedance levels. This improves bandwidth and switching speed. Figure 5-19*b* shows a high-quality active shunt switch for this MDAC. (*University of Arizona*, *Ref. 79; see also* Sec. 5-9.)

at the terminal *Q*, where each bit value a_k is either 0 or 1. This fact is easily proved by the successive Thévenin equivalent circuits shown in Fig. 5-20*b*, *c*, and *d*. Figure 5-20*e* to *f* shows how binary voltage ladders can be connected for conversion of two's-complement-coded positive or negative quantities (see also Fig. 5-18).

Ladder-network DACs necessarily require SPDT switches but need fewer resistors than T-network DACs; above all, binary ladders require only two different resistance values of the same order of magnitude and can, in fact, be made up of resistances *R* or 2*R* alone. This greatly simplifies close matching of film-resistor temperature coefficients; indeed, the entire ladder network can be deposited on a substrate by thick-film or thin-film techniques (Sec. 3-7*b*).

The "inverted" binary ladder shown in Fig. 5-21 is especially suitable for use with the FET SPDT switches of Fig. 5-14*c*. This circuit, which can convert positive or negative digital numbers, is especially useful as an MDAC, while the ladder of Fig. 5-20 is commonly employed in fixed-reference DACs together with bit switches similar to Fig. 5-8*a*.

Ladder DAC networks are readily adapted to conversion of digital codes other than simple binary, and many different circuit combinations are possible. As an example, Fig. 5-22 shows a ladder network suitable for conversion of binary-coded-decimal (BCD) numbers.

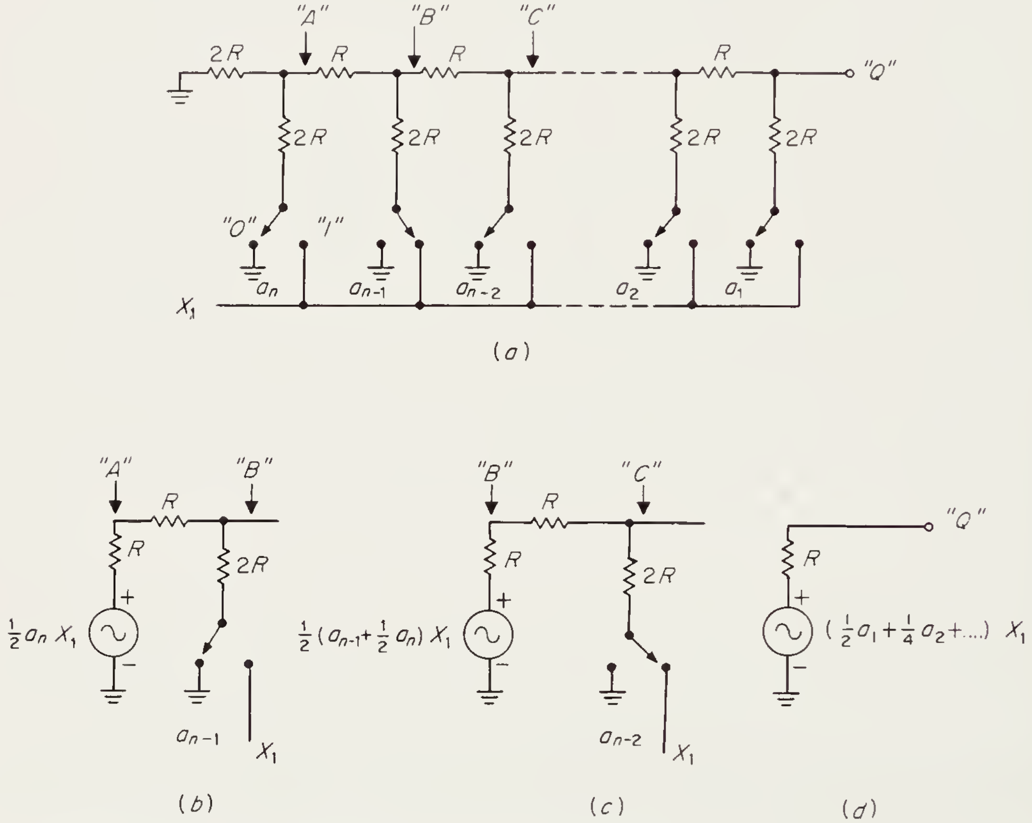


Fig. 5-20*a* to *d*. A binary ladder network (*a*) and (*b*) to (*d*) Thévenin equivalent circuits for successive sets of ladder sections, showing how successive inputs are effectively divided by powers of 2. *Circuit* (*c*) *is the equivalent-voltage-source circuit for the complete ladder.* If such a ladder network is loaded with $2R$, the impedance between each ladder switch and ground is $3R$.

5-8. Current-switching Circuits. The *bipolar-transistor current switches* shown in Fig. 5-23 do not depend on transistor saturation to turn ON. Each switch operates as a feedback-controlled current source in the "working region" of its characteristics when it is ON. To turn the switch OFF, the logic waveform opens an alternate current path through a diode

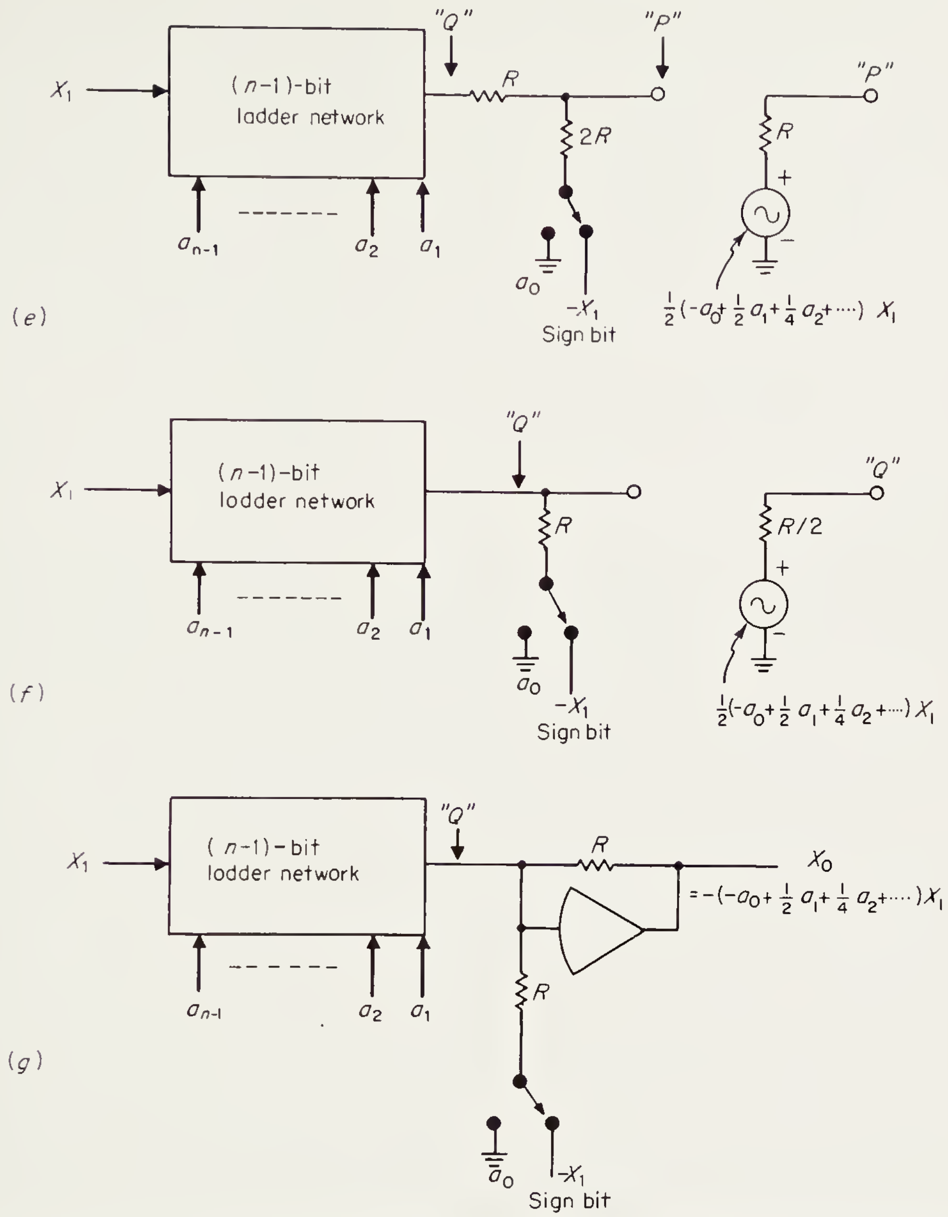


Fig. 5-20*e* to *g*. Ladder circuits for converting two's-complement-coded positive or negative quantities. Equivalent-voltage-source circuits are shown for (*e*) and (*f*).

or transistor ("current stealing") and cuts the current-source transistor OFF. Compared to voltage switching, current switching has two significant advantages:

1. With reasonably high current gain in the current-source feedback circuit, the output current does not depend on switch-transistor forward resistance.
2. Current switches turn OFF quickly, because there is no need to get a transistor out of saturation (see also Secs. 5-5 and 5-17).

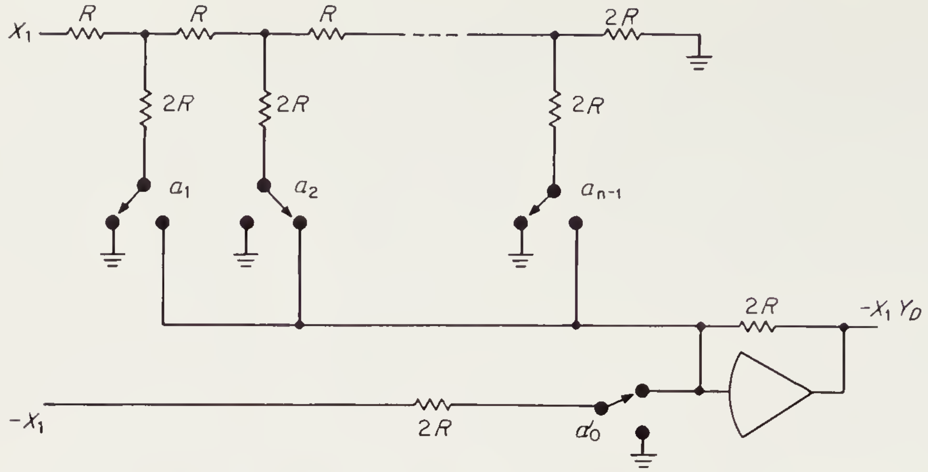


Fig. 5-21. An "inverted" binary ladder suitable for use with the FET/bipolar SPDT switches of Fig. 5-14.

Switched current sources can be used as operational-amplifier inputs in DAC circuits similar to Figs. 5-17 and 5-18, or in *current-ladder circuits* like that shown in Fig. 5-25*a*, which, like binary voltage ladders, employ only two different resistance values. Current switches are employed mainly in fixed-reference DACs, because the inherent high impedances of current sources tend to cause analog-input phase shift.

The output of the simple switched current sources in Fig. 5-23 depends, unfortunately, on transistor V_{BE} and thus on temperature, so that accurate current-switch DACs require some form of temperature compensation. In the ingenious current-switch network of Fig. 5-24, the ON current level in five closely matched transistors is determined by an operational-amplifier feedback circuit. V_{BE} matching is close, since all transistors are deposited on the same monolithic integrated-circuit chip. Transistors designed to pass successive bit currents of 1/8, 1/4, and 1/2 mA are given successively doubled emitter areas, so that similarly good current-gain (β) tracking is obtained. Figure 5-25*b* shows how several 4-bit DAC elements of this type are combined into accurate 8-bit or 12-bit DACs; while only the first 4 bits are obtained from

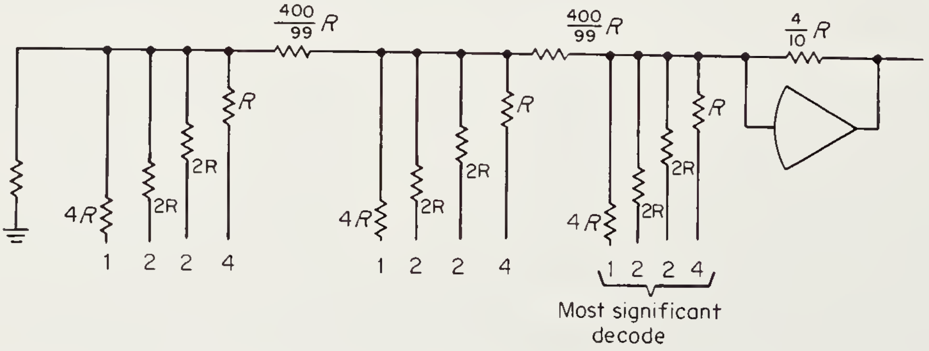


Fig. 5-22. A binary-coded-decimal (BCD) ladder.

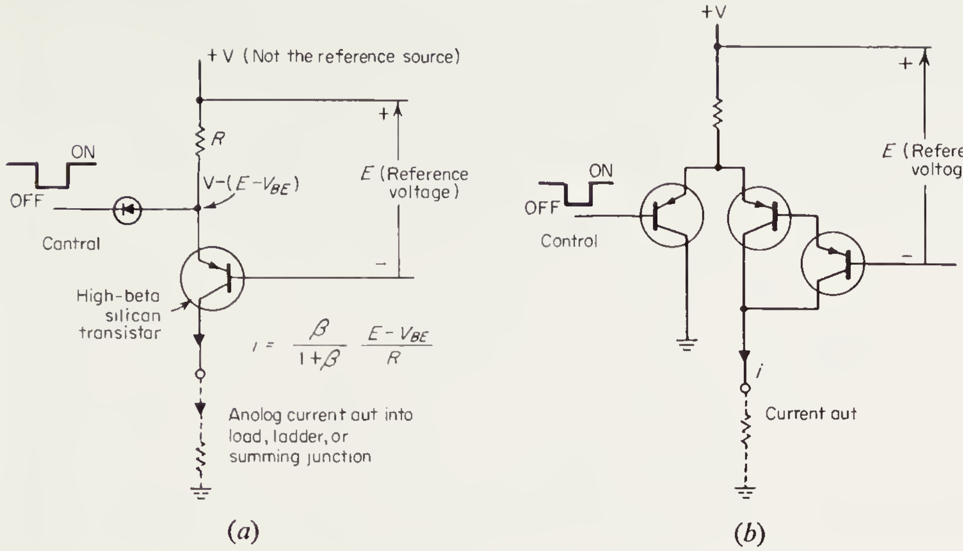

+V (Not the reference source)
R
ON
OFF
Control
V-(E-V_BE)
E (Reference voltage)
+
−
High-beta silicon transistor
i = β/(1+β) · (E - V_BE)/R
Analog current out into load, ladder, or summing junction
(a)
+V
+
E (Reference voltage)
−
ON
OFF
Control
i
Current out
(b)

Fig. 5-23*a*, *b*. Switched current sources suitable for fixed-reference digital-to-analog convectors. Monolithic integrated-circuit construction is possible. The V_{BE} temperature dependence should be compensated in the source of the reference voltage E. Reverse unctions, voltages, and control for negative-current sources.

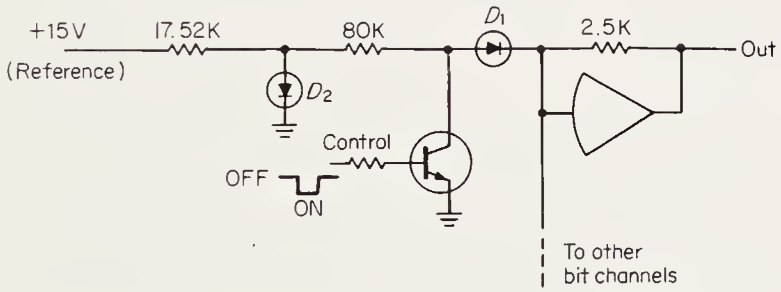


Fig. 5-23*c*. Another type of switched current source used in a fixed-reference DAC. D_2 provides V_{BE} compensation. (*Sprague Electric Co.*)

the same integrated circuit, the less significant bits need not be matched as accurately.

5-9. Errors and Specifications. The *resolution* of a digital-to-analog converter is given by the number of bits. Conversion *accuracy* should be within the voltage limits determined by one-half of the least significant bit (½ LSB), so that the DAC analog output will always increase (or always decrease) with the digital input (*monotonicity*).

DAC static errors, due to *network-resistance tolerances, switch resistances, switch offsets, and reference-voltage errors*, must be figured *on a worst-case basis for all digital inputs, working temperatures, and power-supply-voltage limits.* In general, the sign bit and the most significant bits will contribute relatively larger switch-resistance and switch-offset errors than the less significant bits. Reference 4 contains a number of carefully detailed worst-case-design examples.

As already noted in Sec. 5-1, DAC-network resistances large compared to semiconductor resistances favor static accuracy but increase the *settling time* required for the DAC output to settle within its specified accuracy; *better accuracy will require relatively longer settling times* (Fig. 5-26). For any given accuracy, faster conversion will require lower network impedances, which may have to be matched to individual bit switches at substantial expense. Table 5-1 shows some typical specifications.

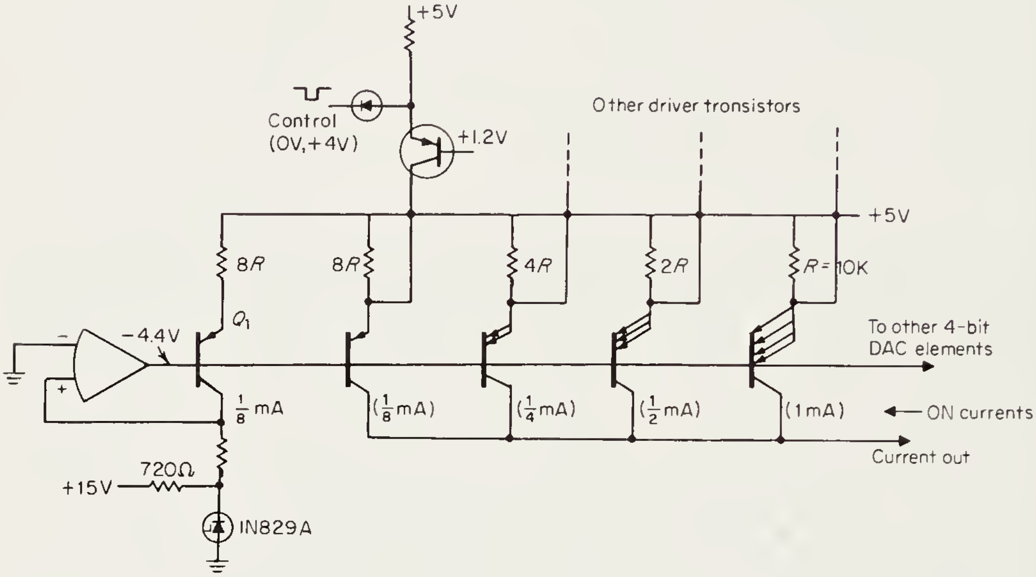


Fig. 5-24. Trimmer-free 4-bit current-switching DAC element for a 12-bit fixed-reference DAC (Fig. 5-25*b*). A feedback loop including the extra current source Q_1 sets all current-source base voltages to compensate for V_{BE} and β changes with temperature and power supply. The four current sources shown correspond to the most significant bits and are deposited on the same substrate as Q_1 for V_{BE} tracking within 500 μV between −50 and +120 deg C. Similarly good β tracking is achieved by doubling the emitter areas of successive current sources for equal current densities. (*Simplified circuit of AD* 550, *Analog Devices, Inc.*)

Analog-input phase shift in MDACs is also improved by low network impedances and can be further reduced by equalizing-capacitor networks in input and/or feedback networks (e.g., as in Fig. 3-17*c*). Figure 5-27 illustrates the operation of a fast MDAC; phase-shift errors within 0.1 percent up to 10 KHz are readily possible.

Finally, DAC specifications include *voltage ranges for analog input and output* and the digital-input logic levels. Many DACs include digital registers and buffer registers (Sec. 7-9*c*).

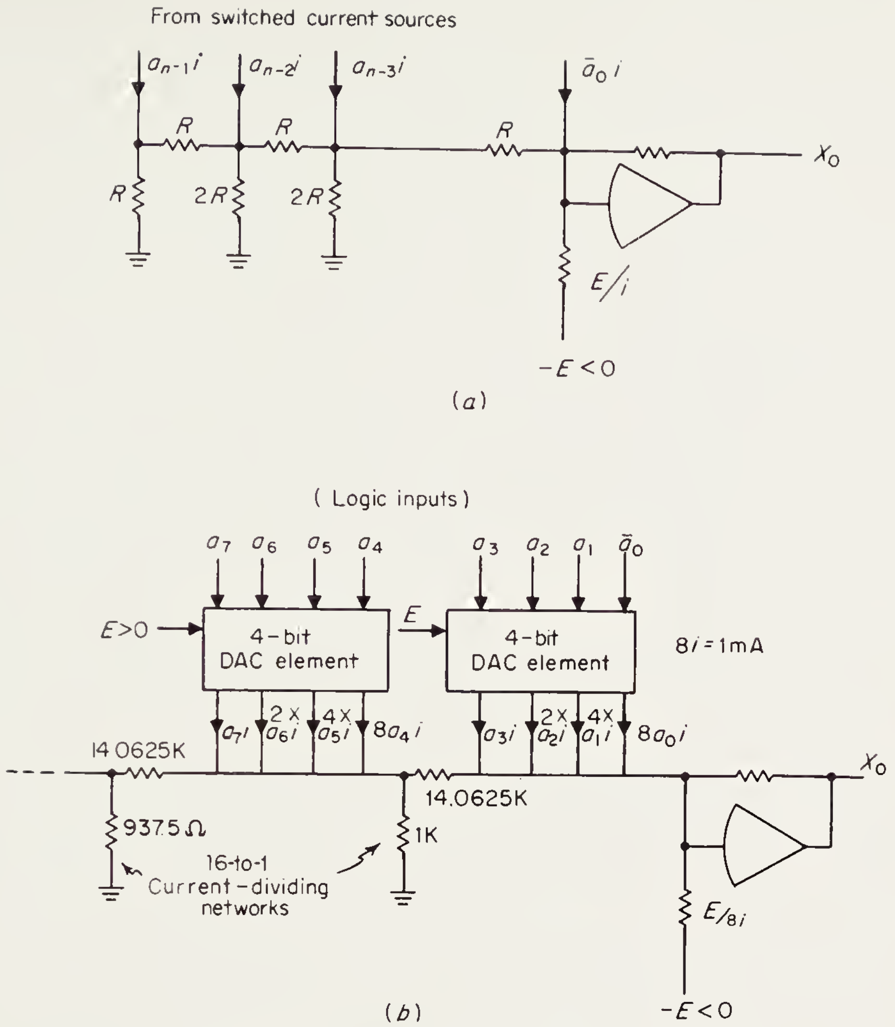


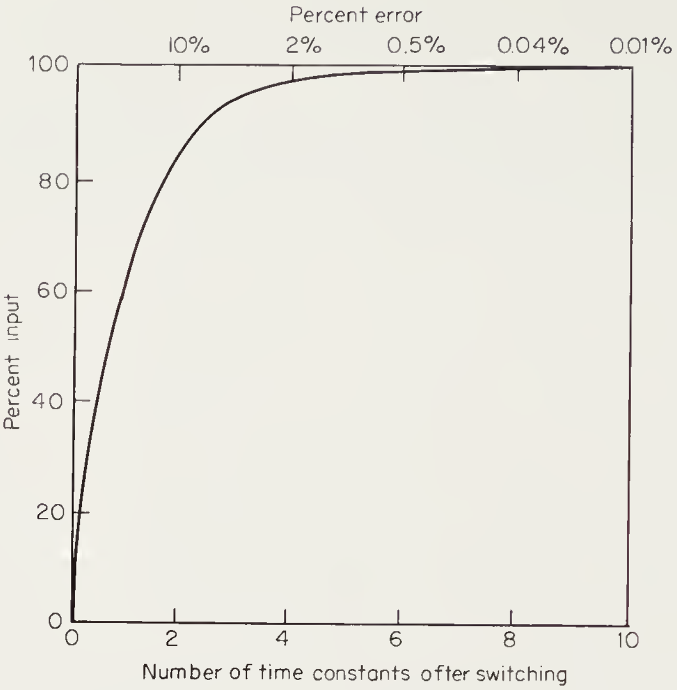
Fig. 5-25. Fixed-reference DACs employing switched current sources. The sign-changing scheme of Fig. 5-25*b* permits the use of similar one-directional current sources. Each a_k is either 0 or 1, and E is a fixed reference voltage.

Table 5-1. Typical Digital-to-analog Converters*

1. *Low-cost:* 8 to 12 bits, temperature range 0 to 50 deg C (or −50 to 150 deg C, more expensive); settling time 0.3 to 10 μsec. Resistances 10K to 500K.
2. *High-quality, designed for slow conversion:* to 15 bits, 0 to 50 deg C; settling time 50 μsec. Fixed-reference DACs for digital voltmeters, precision instruments; MDACs of this type serve as *digital attenuators* in hybrid computers; switch only *between* analog-computer runs. Available for ±100 volts. Resistances 100K to 500K.
3. *Fast, high-quality:* 14 bits, 0 to 50 deg C; settling time 0.5 to 5 μsec. For hybrid-computer applications requiring fast conversion *during* analog-computer runs. Expensive. Available for ±100 volts. Resistances 5K to 20K.
4. *Very fast, low-resolution:* 4 to 8 bits, −50 to 150 deg C, settling times below 0.05 μsec for communications applications. Resistances 1K to 10K.

---

* Accuracy within ½ LSB; settling times to within static accuracy; do not include output-amplifier settling. Output range is ±10 volts unless noted.



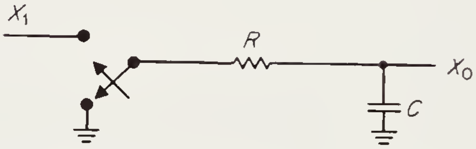


Fig. 5-26. The step response of a simple switched *RC* circuit with time constant *RC* shows typical settling times required for given accuracy.

Switching-spike effects are serious in many DACs. In addition to ordinary switch-capacitance spikes, large if short spurious transients can be caused by *unequal bit-switching times.* Where spike effects are critical (e.g., "glitches" in digitally operated cathode-ray-tube displays), one may use a sample-hold circuit (Sec. 5-17) to hold the old DAC output until the new DAC output has settled after the spike. The sample-hold then switches back into TRACK with (it is hoped) a small switching transient.

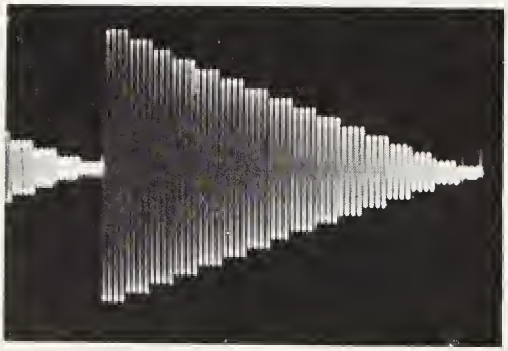


Fig. 5-27. Multiplication of a ±10-volt 200-KHz sine wave by digital words at 50,000 samples/sec.

Whenever, in a digitally controlled voltage

$$X_0 = Y_D X_1$$

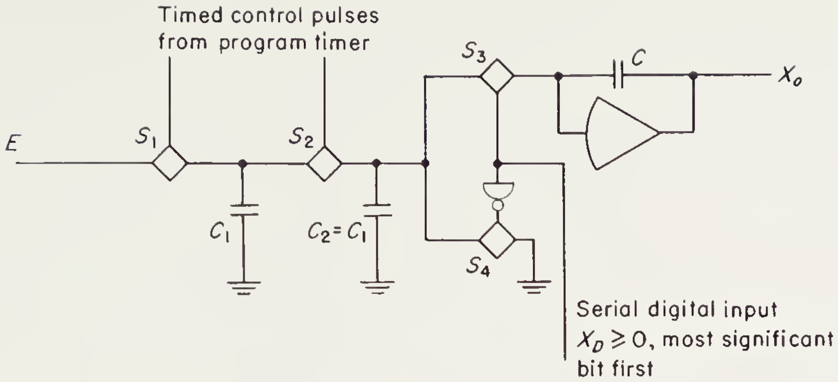


Fig. 5-28. This simplified switched-capacitor decoder accepts unipolar serial input. Switches close *momentarily* to transfer charges in the following sequence:

1. At the start of each sample period, S_1, S_2 open to charge C_1 and $C_2 = C_1$ to E volts.
2. If first (most significant) bit is 1, S_3 discharges C_2 into integrator, and $X_0 = -(C_1/C)E$. If first bit is 0, S_4 discharges C_2 to ground.
3. S_2 opens; C_1 and C_2 are each charged to $E/2$ volts.
4. If second bit is 1, S_3 adds $-(C_1/2C)E$ to X_0. If second bit is 0, S_4 discharges C_2 to ground.
5. S_2 opens; C_1 and C_2 are each charged to $E/4$ volts, etc. 10-bit accuracy is possible.

the digital input Y_D changes only by a fraction of its full range (-1 to $+1$) from some mean value Y_1, we can gain accuracy and/or reduce cost by generating X_0 as a sum of Y_1X_1 and a scaled-down DAC or MDAC output $aZ_DX_1 = (Y_D - Y_1)X_1$ in the manner of Fig. 2-5.

ELECTRONIC ANALOG-TO-DIGITAL CONVERTERS

5-10. Introduction. *Analog-to-digital converters* (*ADCs*, *encoders*) produce a digital output proportional to an analog input X_A voltage or (usually) to the ratio of an analog input X_A and an internal or externally supplied *reference voltage E.* As a rule, each conversion is started and timed by a *convert command* in the form of a pulse or level change supplied by a switch, digital computer, timer, or sensing device. Each conversion will require a finite time, during which the input may or may not have to be "held" by a sample-hold circuit (Sec. 5-17). End of conversion is usually indicated by a level change in the output of a flip-flop (DONE flag, Sec. 7-10).

The basic ingredients of all electronic ADCs are *analog comparators* (Secs. 4-9*a* and *b*), whose binary output indicates the sign of an analog input sum or difference.

5-11. Specifications. ADC specifications are similar to DAC specifications and include *resolution* (number of bits), *accuracy* over stated

Table 5-2. Binary-number Representation of Voltages

Unsigned octal numbers*	Voltage
0000	0
0001	0.00244140625
0002	0.0048828125
0004	0.009765625
0010	0.01953125
0020	0.0390625
0040	0.078125
0100	0.15625
0200	0.3125
0400	0.625
1000	1.25
2000	2.5
4000	5
6000	7.5
7000	8.75
7400	9.375
7600	9.6875
7700	9.84375
7740	9.921875
7760	9.9609375
7770	9.98046875
7774	9.990234375
7776	9.9951171875
7777	9.99755859375
10000	10

* Each octal digit may be regarded as *a shorthand notation for three binary digits*, thus:

Octal	Binary	Octal	Binary
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

temperature and power-supply-voltage ranges (usually within ½ LSB), and *conversion time or rate.* In addition, one must specify the input voltage range and impedance, reference-supply requirements, and the digital-output code and logic levels.

5-12. Ramp/Comparator and Integrating Converters. (a) Ramp/Comparator Converters. A simple *ramp/comparator converter* (*analog-to-time-to-digital converter*) gates clock pulses into a counter while an accurate ramp voltage varies between zero and the analog input $X_A > 0$ (Fig. 5-29*a*). The counter, which serves as the ADC output register, is then read and reset to zero. The second comparator response can initiate a DONE (conversion completed) signal.

Either binary or decimal counters can be used. To read positive or negative voltages between -10 and $+10$ volts, we can set the first comparator in Fig. 5-29*a* at -10 volts instead of 0 volts and employ a binary (two's-complement) counter initially reset to -10_{10} instead of 0. The converter will then read the correct two's-complement code (see also Fig. 5-29*b*).

Ramp/comparator converters require that the analog input remain constant after a timed conversion command, so that a sample-hold circuit may be needed. The ramp/comparator converter circuit of Fig. 5-29*b* employs an existing track hold integrator of a fast analog computer as both a sample-hold *and* a ramp generator.[45]

(b) Integrating Converters. If a constant or variable voltage $X_A > 0$ (one could handle negative inputs by adding a known positive offset) is applied to the input of an electronic integrator initially reset to zero, the integrator output will take ERC/X_A sec to reach the negative reference level $-E$. In *analog-to-frequency-converter ADCs* the integrator is then quickly reset to zero, and the process repeats (Fig. 4-30*a*). The average frequency of the reset pulses, measured by the pulse count during a specified *observation period T*, is approximately

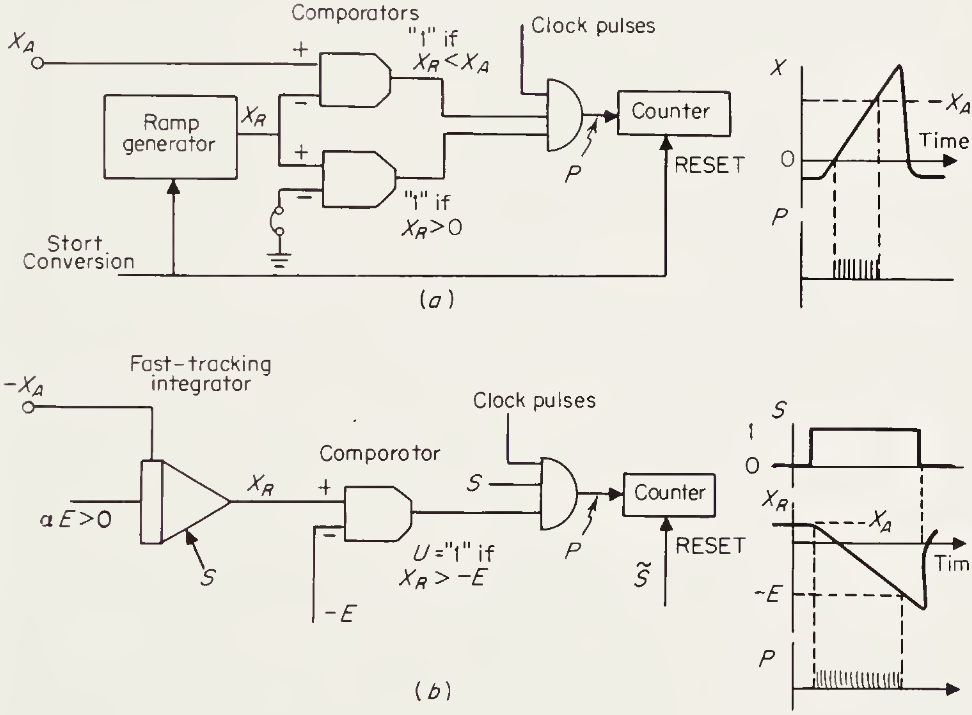


Fig. 5-29. Ramp/comparator converters.

$$\frac{N}{T} \approx \frac{1}{ERC}\bar{X}_A \qquad \bar{X}_A = \frac{1}{T}\int_0^T X_A(t)\,dt \tag{5-5}$$

where we have neglected the effect of the integrator-resetting time. This will be permissible if T_{RESET} is less than the desired measurement-error percentage of the integrator time constant RC, and $X_A \leq E$. In practice, T_{RESET} can be less than 10 μsec, so that $RC = 10$ msec is large enough for 0.1 percent error (some of the reset-time error could be "calibrated out").

Reset-time errors could be reduced by employing two integrators, one integrating while the other one resets. A better voltage-to-frequency conversion scheme switches the input of a single integrator between X_A and $-X_A$ whenever the integrator output reaches $-E$ and $+E$. This method does away with reset switching and also compensates integrator d-c offsets.

The *dual-slope integrating ADC* shown in Fig. 5-30*b* starts with its integrator output at 0 volts and then integrates the analog input $-X_A < 0$ for N_R clock-pulse periods or $T = N_R\,\Delta t$ sec. Digital logic now switches the integrator input to a $+E$ reference, and comparator/logic circuits count N pulses while the integrator output returns to 0 volts. Then

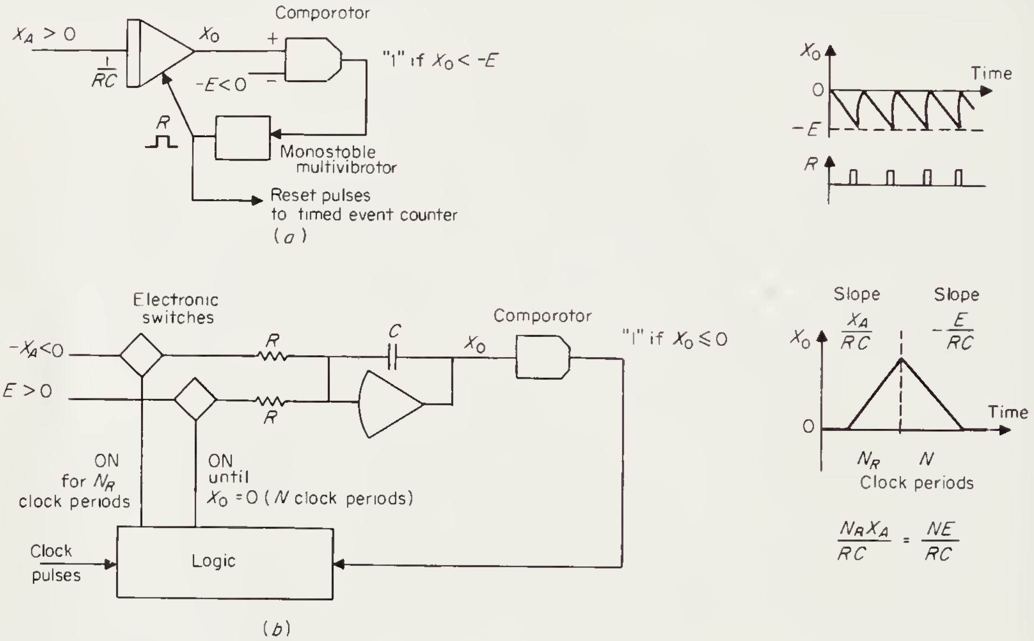


Fig. 5-30. Analog-to-frequency converter (*a*) and dual-slope converter (*b*).

$$N = \frac{N_R}{E}\,\bar{X}_A \qquad \bar{X}_A = \frac{1}{T}\int_0^T X_A(t)\,dt \tag{5-6}$$

which is independent of the integrator RC and also of the clock frequency, so that no precision crystal oscillator is needed. Dual-slope ADCs serve both in inexpensive and precision (0.01 percent of half-scale ±½ LSB) digital voltmeters, with conversion times between 0.1 and

50 msec. References 67, 70, and 73 describe practical circuits, including bipolar operation through input biasing, and use of vernier counters.

Integrating ADCs read the *time average* $\bar{X}_A$ of a possibly variable analog input X_A over a specified observation time T. *This is especially useful for rejecting periodic and nonperiodic high-frequency noise in low-level d-c measurements.* In particular, line-frequency-noise effects can be reduced by at least 70 db if we make the observation interval T a multiple of the line-frequency period. It may also be possible to "synchronize" the observation time with other periodic noise. For *timed* measurements, integrating ADCs will, again, require sample-hold circuits.

5-13. Feedback-type Analog-to-digital Converters. (a) General Principle. To obtain high accuracy and fast conversion independent of capacitor properties, we use a digital-to-analog converter to reconvert our digital output X_D to a voltage EX_D, which is compared with the analog input X_A (Fig. 5-31*a*); E is the ADC reference voltage. With $|X_A| \leq E$ and $|X_D| \leq 1$, we vary X_D during each conversion cycle until

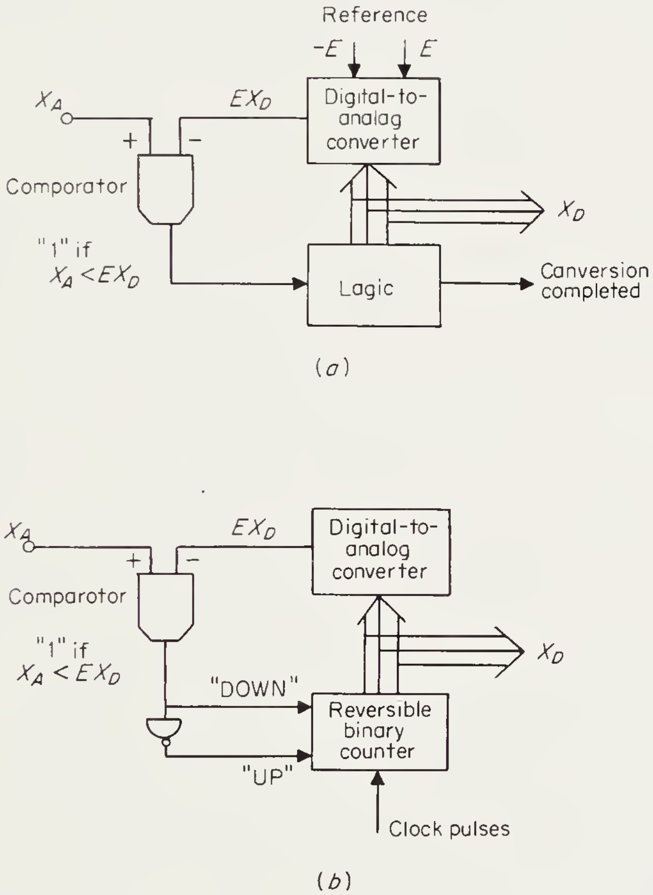


Fig. 5-31. Principle of a feedback-type ADC (*a*) and "continuous" (reversible-counter-feedback) converter (*b*).

the magnitude of the *analog error* $e_A = EX_D - X_A$ is sufficiently small (i.e., less than ½LSB):

$$X_D = \frac{X_A + e_A}{E} \qquad |e_A| < \frac{E}{2^n} \tag{5-7}$$

where n is the required number of bits, *including the sign bit*. Different schemes for varying X_D yield various compromises between circuit simplicity and maximum n-bit conversion time. We shall measure conversion time in *clock periods*, where each clock period must be sufficiently long to permit comparator, logic, and D/A converter switches to settle. Typical clock periods will be between 0.05 and 10 μsec.

In the simplest scheme, the D/A converter register is a clock-driven 2^n-step counter which starts at the most negative value once per conversion cycle and counts up to produce a staircase D/A output analogous to the ramp voltage in Fig. 5-29*a*. X_D is read when the staircase voltage reaches X_A. *The maximum conversion time is* 2^n *clock periods*, since each staircase step will be just $2E/2^n$ volts.

(b) "Continuous" Converters. To reduce the average conversion time, the D/A converter register can be a reversible counter; we count up or down as required by the sign of the error $e_A = EX_D - X_A$ (*continuous* or *incremental converter*, *digital servo*, Fig. 5-31*b*). A small comparator deadspace about $e_A = 0$ will prevent oscillation about the correct output. Incremental converters save appreciable time if X_A is not a random input (like a multiplexer output) but a time-variable analog voltage *known to increase and decrease no faster than* $R < 2E/2^n$ *volts per clock period*. In this case, the first conversion can take up to $2^nR/2E$ clock periods; the converter will then *track the signal* in successive clock periods. We can track faster signals at the expense of circuit complexity if we add comparators which sense large analog errors and transfer the counter input to a higher digit position (variable-increment counter).

(c) Successive-approximation Converters. For faster conversion, we program the D/A converter register with digital logic to "weigh" X_A by successive binary-digit approximations. We start with $X_D = 0$ to find the sign of X_A. If $X_A > 0$, we next try $EX_D = E/2$ to see if the most significant binary digit is 0 or 1, and continue on to the last binary digit (*successive-approximation converter*). *The required conversion time will be n clock periods, one for each bit in* X_D.

Figure 5-32 illustrates the operation of a typical successive-approximation converter. The digital output X_D, and thus the DAC output voltage EX_D, is set by the *ADC* (*output*) *register*, comprising n flip-flops $F0, F1, F2, \ldots$. Successive X_D values will be programmed by a digital 1 passing through an $(n + 1)$-bit *shift counter* (flip-flops $S0, S1, S2, \ldots$).

The START CONVERSION pulse resets the shift counter to 1000 . . . and the ADC register to 0000 . . . ; this also releases the test pulses and shift

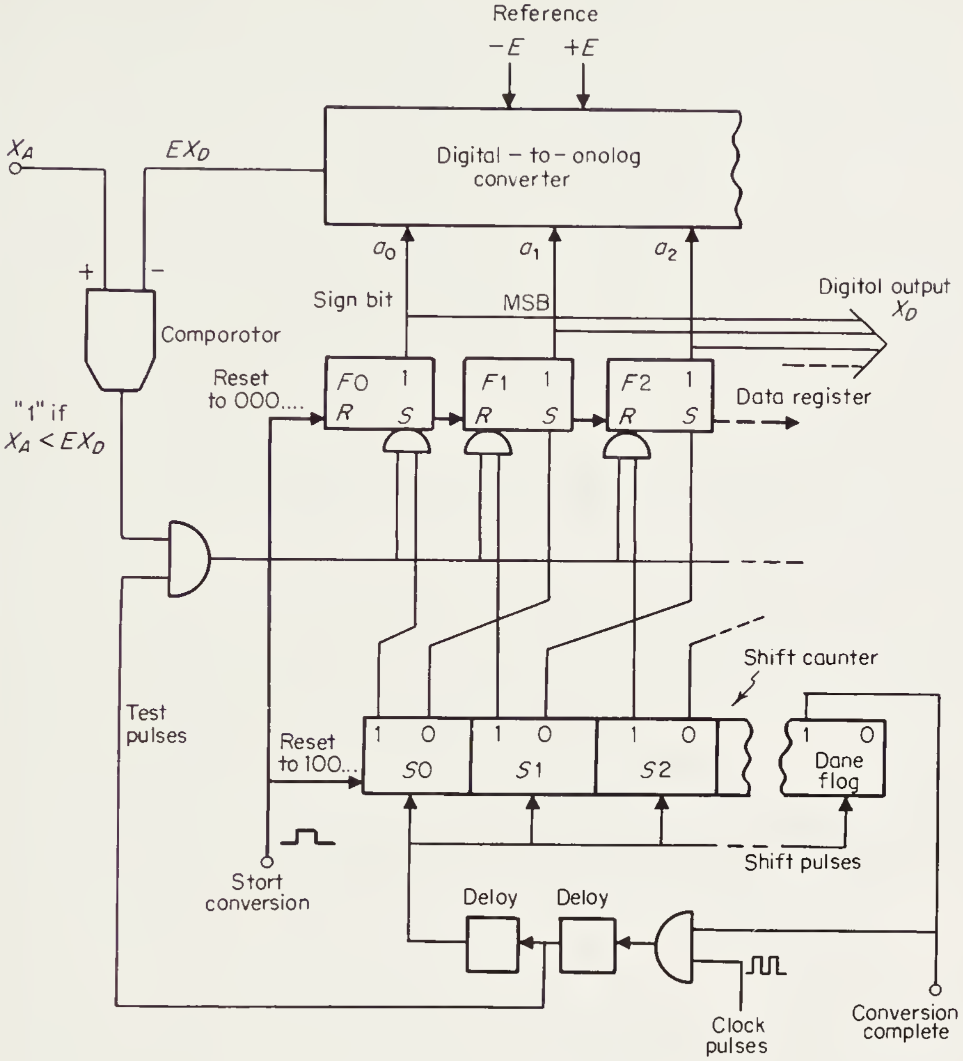


Fig. 5-32. Typical successive-approximation ADC.

pulses produced by the ADC clock. Initially, $X_D = 0000\ldots.$ If now $X < EX_D = 0$, the comparator output will be 1, and the first test pulse (suitably delayed to let flip-flops, DAC, and comparator settle) will set the sign-bit flip-flop $F0$ to 1. The first shift pulse follows and shifts the counter to $0100\ldots.$ We then have

$$X_D = \begin{cases} 1100\ldots & \text{if } X_A < 0 \\ 0100\ldots & \text{if } X_A \geq 0 \end{cases}$$

and the corresponding DAC output ($-E/2$ or $E/2$) is compared with the analog input X_A. If (and only if) $X_A < -E/2$ or $0 \leq X_A < E/2$, the second test pulse will correctly reset the MSB flip-flop $F1$ to 0. Otherwise, the most significant bit remains, as it should, at 1. Successively following shift pulses and test pulses similarly determine the other bits of X_D in order of decreasing significance. The nth shift pulse shifts the 1

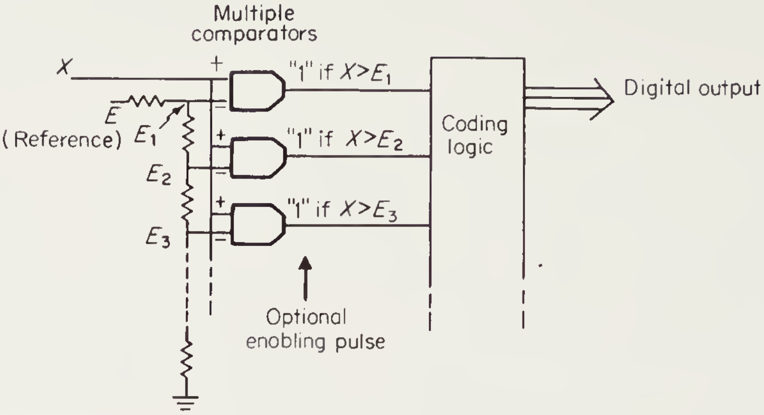


Fig. 5-33*a*. "Instantaneous" converter employing parallel comparators. A timed strobing pulse can sample the input either by enabling the comparators or by gating the digital output.

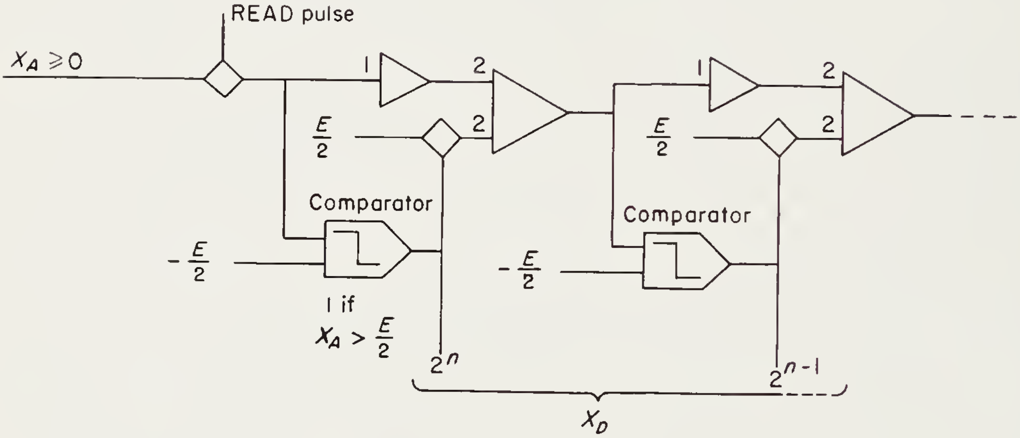


Fig. 5-33*b*. A cascade encoder.

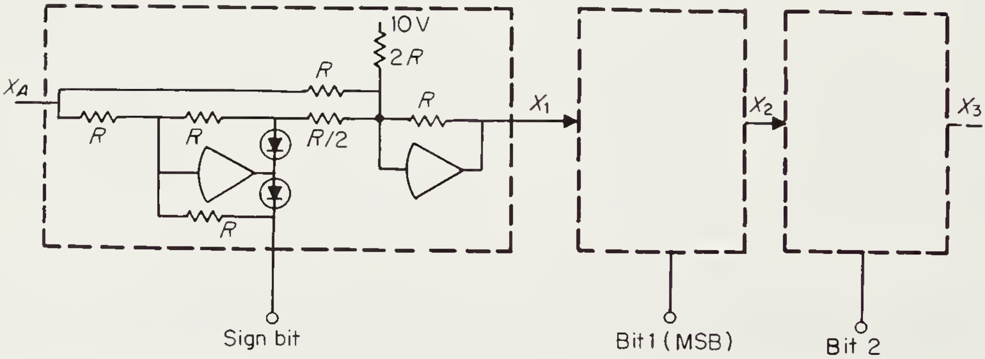


Fig. 5-33*c*. Principle of a cascade encoder employing precision-limiter absolute-value circuits. In practice, the summing amplifiers between stages can be replaced by summing-network addition.

into the $(n + 1)$st shift-counter flip-flop, which serves as a DONE flag signaling completion of the conversion.

- If the number of ADC bits is larger than 8, it may pay to replace the $n + 1$ shift-counter flip-flops by a binary counter with $N + 1$ decoding gates (Appendix B), since this binary counter would require at most $\log_2 (n + 1)$ flip-flops.[4]

5-14. Converters Employing Multiple Comparators. Converters using parallel comparators for each quantization level (Fig. 5-33*a*) permit the highest possible conversion speed, since all digits are produced simultaneously; 3- to 4-bit converters of this type are available as monolithic integrated circuits.

A *cascade encoder*[75] reduces the required number of comparators but produces binary digits in succession. Referring to Fig. 5-33*b* ($X_A > 0$), comparator 1 decides whether $X_A > E/2$. If $X_A > E/2$, comparator 2 decides whether $X_A > E/2 + E/4$; if $X_A < E/2$, comparator 2 decides whether $X_A > E/4$, etc. Cascade encoders can be very fast. The converter shown in Fig. 5-33*b* requires one clock period per bit, but improved circuits separate the converter stages with analog delay lines one clock period (0.1 μsec) long, so that the earlier stages can begin to work on the *next* analog sample before each conversion is completed.

Figure 5-33*c* shows a ±10-volt cascade encoder based on the precision absolute-value circuit of Fig. 4-23.[76, 77] The *first encoder stage* produces a digital 1 if $X_A < 0$ (sign bit) and also an analog output X_1 such that

$$X_1 > 5 \text{ volts if } \begin{cases} X_A > 5 \text{ volts} & (X_A > 0) \\ X_A < -5 \text{ volts} & (X_A < 0) \end{cases}$$

The *second encoder stage*, quite similar to the first, produces a digital 1 if $X_1 - 5 \text{ volts} > 0$, and also a new analog output X_2 to be tested by the *third encoder stage*, and so on. The conversion time will be the time required for the signal to propagate through n stages, typically 0.2 μsec per stage with fast operational amplifiers. Note that the code produced, is not the usual binary code, but a "reflected" code (*Gray code*), which requires code conversion.[76] Many variations of this circuit exist.[77]

Cascade encoders are usually limited to about 8 bits because of error accumulation in successive stages.

5-15. Discussion. Accuracies and Conversion Rates. "Instantaneous" (parallel-comparator) ADCs can be very fast, permitting conversion rates of at least 10^8 words/sec, if one is satisfied with 4- to 6-bit resolution. Such fast but coarse converters are useful for pulse-code-modulation communications and can be fabricated as monolithic integrated circuits. *Cascade encoders* are next in speed. Their resolution is usually limited to 8 or 9 bits, with 10^7 bits or about 10^6 words per second.

The conversion rate of a *ramp/comparator* or *integrating ADC* is limited by analog-switching and comparator-response times, whose unpredictable fluctuations must stay within the specified error percentage of the total conversion time; 7 to 8-bit ramp/comparator circuits used in special instrumentation[71,72] can convert at up to 10^6 words/sec. Digital-voltmeter designers, who rarely require more than 1,000 words/sec, prize *integrating converters* for their noise rejection (Sec. 5-12*b*). Accuracy of any ADC using a capacitor for ramp generation or signal integration is limited by dielectric absorption (Sec. 3-9) to within 0.01 to 0.02 percent of half-scale ±½LSB (about half the dielectric-absorption effects can be calibrated out). 11-bit integrating converters can convert at up to 5,000 words/sec.[73]

The best accuracies, as well as high conversion rates at extra cost, are obtained with *successive-approximation converters*, which are predominant in hybrid computation. Inexpensive ADCs of this type yield 8 to 12 bits at 1 to 2 μsec/bit (i.e., of the order of 10^5 words/sec) for less than $300. Up to 15 bits, together with conversion times between 0.1 and 2 μsec/bit, can be obtained at higher prices. The DAC used in any feedback-type ADC is subject to the tradeoff between accuracy, settling time, and cost already discussed in Sec. 5-9. Substitution of multiple slower (and thus less expensive) ADCs, possibly including integrating and "continuous" converters, for the usual combination of multiplexer and high-speed ADC may merit consideration.[74] Other compromises include the design of *subranging ADCs*, which combine a fast "instantaneous" or cascade ADC having only a few bits with a slower, more accurate ADC of the successive-approximation type.[68]

INTEGRATOR MODE CONTROL AND TRACK-HOLD CIRCUITS

5-16. Electronic Mode-control Switching. The very simplest scheme for resetting an integrator merely resets it to zero by shorting the integrating capacitor (say with a JFET switch); the desired initial output is then added to the output of the integrator proper. In spite of its simplicity, this resetting method is rarely used, because the two voltages to be added must be scaled down and thus lose accuracy.

Relay switching in the basic three-mode (RESET-COMPUTE-HOLD) integrator circuit of Fig. 1-10 is now obsolete in view of *the critical timing needed when switching into* COMPUTE (Fig. 3-2). Most general-purpose analog computers employ FET/bipolar-transistor switching similar to Fig. 5-34. The simple HOLD switching scheme of Fig. 1-10 would require a possibly inaccurate series switch between the integrator summing junction *SJ* and the amplifier input *G*; this is avoided (at considerable expense) by the separately switched HOLD capacitor in Fig. 5-34, where now all MOSFET forward resistances are harmlessly inside feedback loops in the manner of Fig. 5-15*b*.

The two-mode (RESET-COMPUTE) integrator of Fig. 5-35, designed for an ultrafast iterative analog computer with low-impedance computing networks (Sec. 7-2*b*), employs only a single, very fast switch. This switch is a switched unity-gain follower, which amplifies the RESET current, effectively; the integrator sees reset-network impedances of the order of the follower output impedance. In the RESET mode (Fig. 5-35*b*), the integrator output will settle to the value

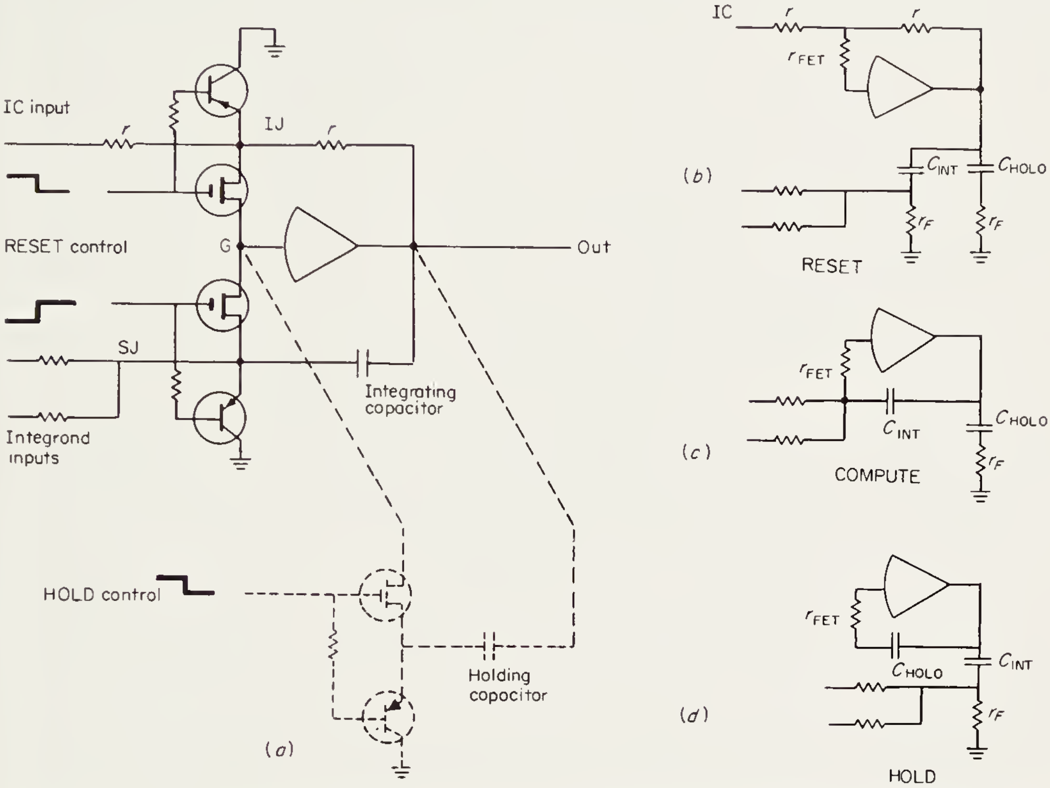


Fig. 5-34. Three-mode integrator-control circuit. Typical integrators of this type can hold within 0.01 percent of half-scale (10 volts or 100 volts) for at least 200 sec/μF and switch into COMPUTE or HOLD within 200 to 400 nsec, with at most 50 to 100 nsec between integrators.

$$X_0 = -(X_{T1} + X_{T2}) - 0.2\left(\frac{X_1}{1{,}000} + \frac{X_2}{10{,}000}\right)$$

We see that, although the integrand inputs are never switched off, even X_1 can affect the RESET output only by at most 0.02 percent of half-scale. The design accuracy of the integrator is 0.1 percent of half-scale. This circuit also serves as a fast track-hold (Sec. 5-17*c*), which is the reason for having two RESET/TRACK inputs.

The maximum time needed to reset the electronic integrators of

Figs. 5-34 and 5-35 within stated accuracy limits is determined not so much by their capacitor-charging time constants $C_{\mathrm{INT}}r_F$, $C_{\mathrm{HOLD}}r_F$ sec (Fig. 5-34*b*) and $0.2C$ sec (Fig. 5-35*b*) as by the rate at which amplifiers and switches can supply the charging current (see also Sec. 5-17).

5-17. Track-hold Circuits (Sample-hold Circuits). (a) Basic Track-hold Circuit. Performance Specifications. Track-hold circuits (sample-hold circuits) hold a timed value of a variable analog input for analog-to-digital conversion and/or for iterative analog computation (analog "point storage," Sec. 1-13*c*). The output X_0 of the *basic capacitor-storage track-hold* shown in Fig. 5-36 tracks the input X_1 while the switch is ON (TRACK *mode*). A HOLD command opens the switch, so that the capacitor stores the last input value, and the output ideally

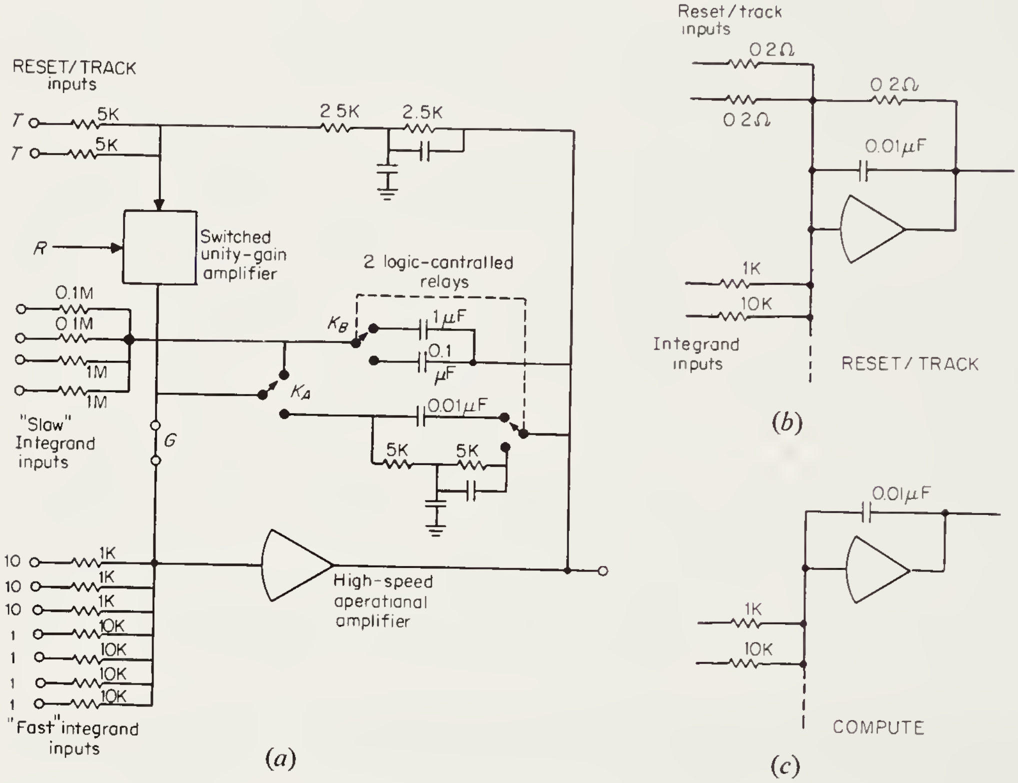


Fig. 5-35*a* to *c*. Two-mode integrator/summer circuit for an analog/hybrid computer permitting very fast iterative operation (up to 4,000 runs/sec). Digitally switched relays permit rate gains as high as $(RC)^{-1} = 10^5$ sec^{-1} with the low-impedance integrator inputs, and a second integrator-input network also permits "slow" operation without excessively large capacitors. In the RESET/TRACK mode (*b*), the switched unity-gain amplifier acts as an impedance transformer, reducing the effective reset/track-network resistances to 0.2 ohms, so that the effect of the 1 or 10K integrator inputs becomes negligible. Typical integrators of this type can hold within 0.1 percent of half-scale (10 volts) for at least 100 sec/μF and switch into COMPUTE or HOLD within 20 to 40 nsec, with at most 5 to 10 nsec between integrators. (*University of Arizona, Refs.* 40, 43, *and* 45.)

remains constant (HOLD *mode*). The storage circuit can drive external loads without discharging their holding capacitors.

An ideal track-hold circuit would follow its input instantaneously and exactly in TRACK, and would hold instantaneously and exactly for an indefinite time after receiving a HOLD command. Figure 5-37*a* illustrates the response of an actual track-hold circuit. The *current capacity of amplifiers and switches* not only limits the maximum tracking rate, but also places a lower limit on the TRACK period required to catch up with

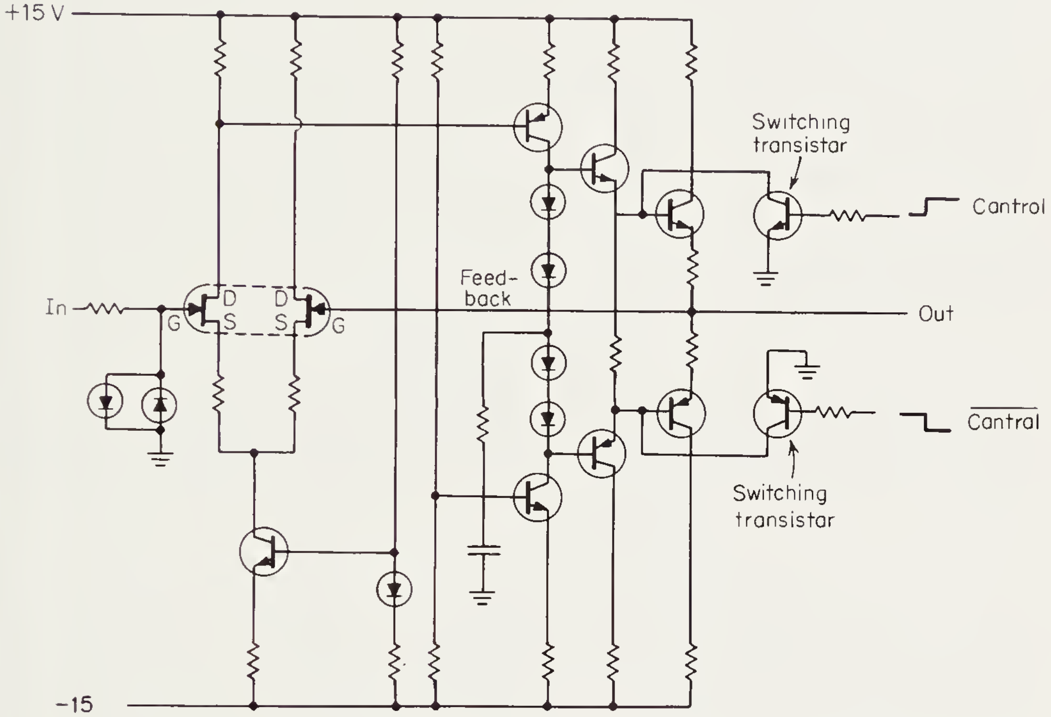


Fig. 5-35*d*. FET-input switched unity-gain amplifier for the integrator of Fig. 5-35*a*. Output impedance is 0.1 ohm at 100 Hz, and output current is 30 mA. The integrator switches into COMPUTE/HOLD within 40 nsec, with ±10-nsec maximum difference between two integrators. Output impedance is 0.1 ohm at 100 Hz; output current is 30 mA. COMPUTE/HOLD drift in 10 msec (0.01 μF) is 2 mV ± 0.05 mV/deg C. In RESET/TRACK with 0.01 μF, the integrator tracks ±10-volt sine waves up to 40 KHz and follows a 10-volt step within 0.1 percent in 10 μsec; small-signal dynamic error is within 0.1 percent up to 10 KHz (no load), and temperature drift is ±100 μV/deg C. (*University of Arizona and Burr-Brown Research Corporation, Tucson, Ariz.*)

specified input-signal excursions (Fig. 5-37*b*). For accurate tracking of high-frequency signal components too small to cause rate limiting, we require low phase shift in TRACK, i.e., sufficient bandwidth in the capacitor-charging input circuit or amplifier. The effect of the capacitor-charging time constant on small-signal phase shift can be compensated with suitable equalizing networks in the input amplifier. The charging time constant must, in any case, be less than one-seventh

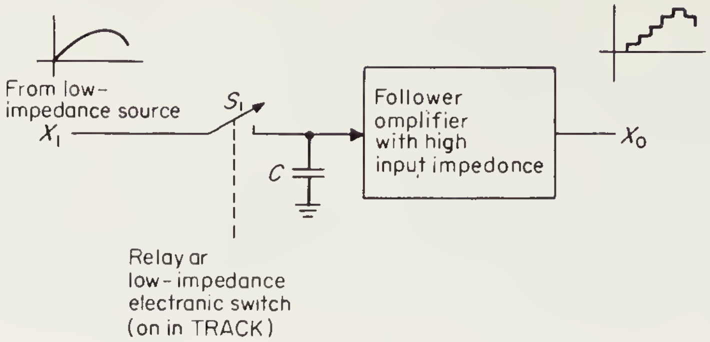

From low-impedance source
X_1
S_1
C
Follower amplifier with high input impedance
X_0
Relay or low-impedance electronic switch (on in TRACK)

Fig. 5-36. Basic switched-capacitor sample-hold circuit.

of the shortest contemplated TRACK period for 0.1 percent accuracy (about one-tenth for 0.01 percent) to allow for exponential step response.

The ideally constant HOLD output *drifts* because of capacitor leakage and dielectric absorption (Sec. 3-9), switch and amplifier-input leakage, and amplifier voltage offset. A total error current i_E contributes the absolute drift error

$$e_H = \frac{|i_E|}{C}\tau_H \tag{5-8a}$$

where C is the holding capacitance, and τ_H is the elapsed time in HOLD. An amplifier offset of e_D volts referred to the input contributes an output

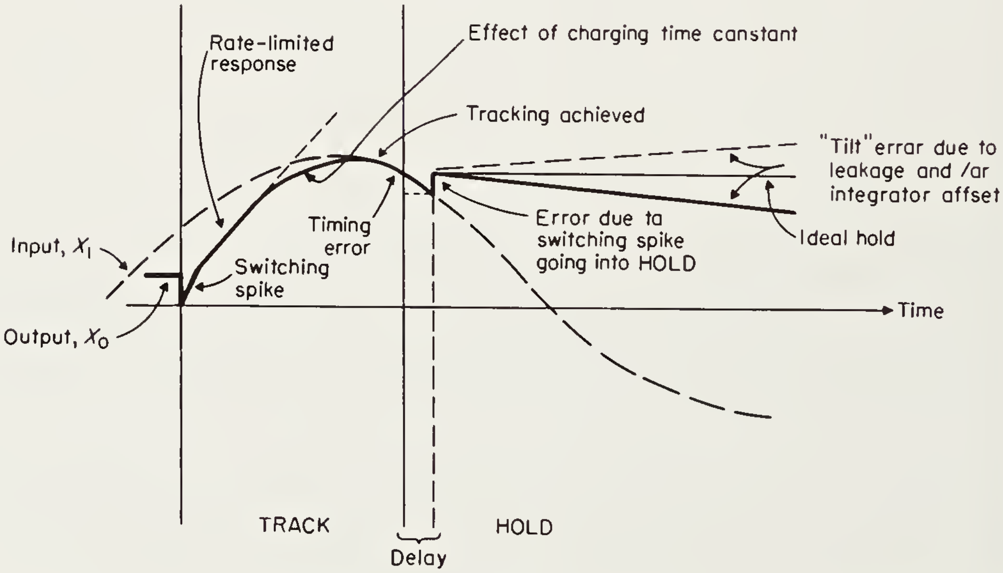


Fig. 5-37*a*. Sample-hold response and error effects. Rate limits are determined by current capacities of switches and/or amplifiers and may differ for positive and negative rates. Switching spikes (for electronic switches only) can be positive or negative. Timing error and switching spike going into TRACK will not affect HOLD output. Errors due to d-c offset and phase shift in TRACK are not shown.

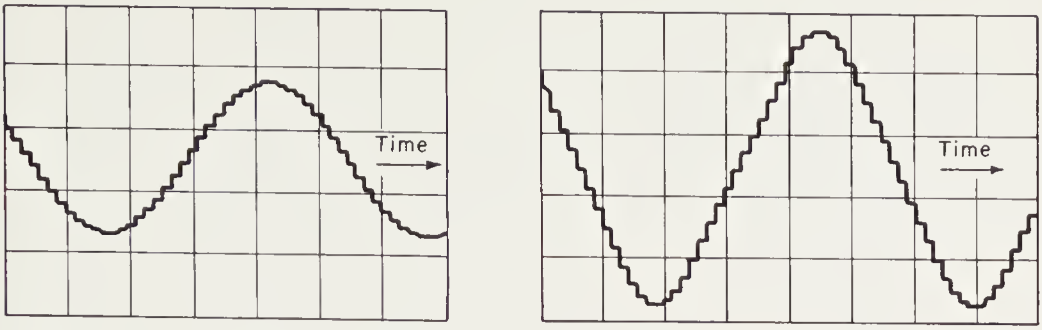


Fig. 5-37*b*. Response of a sample-hold circuit to sine waves at two different amplitudes. The rate limiting is due to a TRACK period too short in relation to the maximum tracking rate.

error e_D in Fig. 5-36, and an absolute output drift

$$e'_H = \frac{|e_D|}{RC}\tau_H \tag{5-8b}$$

in Fig. 5-38*c*, where R is the resistance from summing point to ground in the HOLD mode (see also Sec. 3-21).

With electronic switching, the *switching spike* caused by differentiation of the HOLD control step in the switch capacitances charges the holding capacitor and contributes a spurious voltage step (Fig. 5-37*a*). Its amplitude is approximately

$$e_S = a_S E_C \frac{C_S}{C} \tag{5-9}$$

where E_C is the control-step amplitude and C_S is the effective switch capacitance. a_S is a proportionality factor, which we attempt to reduce through push-pull spike cancellation in symmetrical switching circuits (Fig. 5-4), and/or by adding inverted spikes through a trimmer capacitor (Fig. 5-14*a*). Nonlinearity of electronic-switch capacitances makes accurate spike cancellation for all input-voltage values impractical, but considerable improvement is possible. With well-designed circuits, switching-spike errors are not serious in ordinary track-hold operation, but they may accumulate in certain computations (Secs. 2-5 and 2-6).

Finally, switch- or relay-response delays in executing the HOLD command result in possibly serious *timing errors*, i.e., the sample-hold circuit actually holds a later input voltage value than desired (see also Sec. 3-2*c*).

The holding errors (8) and (9) are seen to improve with increasing holding capacitance C, while low values of C require less current and favor fast tracking. *The required compromise is best resolved with high-current solid-state amplifiers and switches, which permit the use of relatively large holding capacitances.* In practice, i_D and e_D can be kept below 1 to 10 nA and 50 to 500 μV, respectively.

If one must track fast signals accurately *and* one also requires

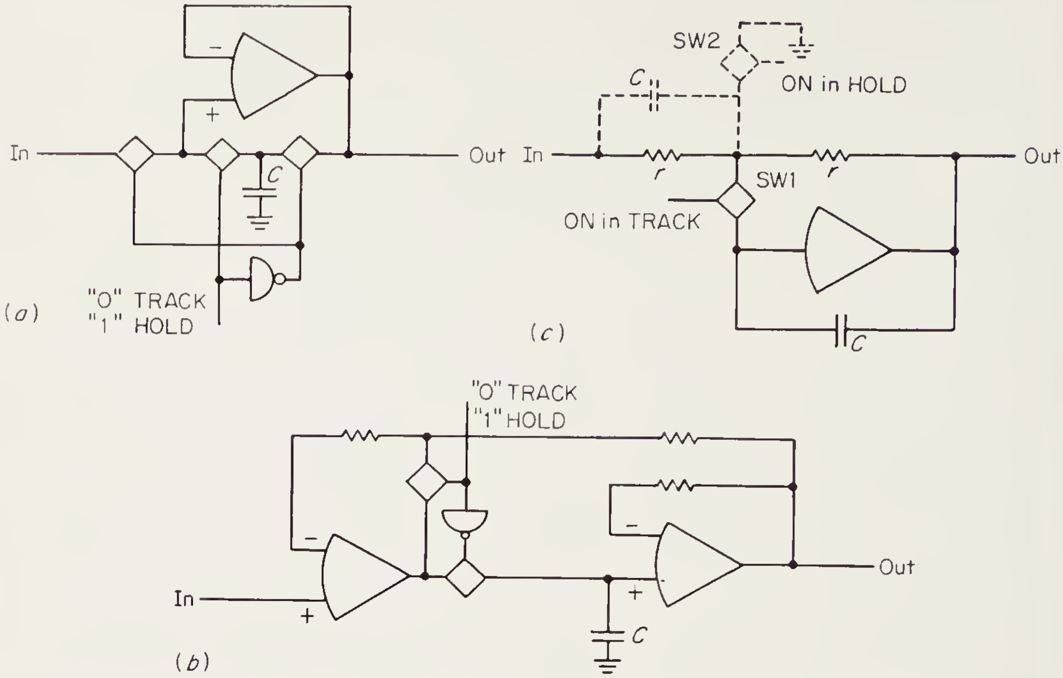


Fig. 5-38. Track-hold circuits. Circuit (*a*) has been implemented with Siliconix, Inc. integrated-circuit FET/drivers.

relatively long holding periods, it is best to employ a fast (low-*C*) sample-hold to track the signal and to sample the output of this fast sample-hold with a second, accurately holding (high-*C*) sample-hold after the input sample-hold has just settled into HOLD. Such *track-hold* pairs are, in any case, needed in applications requiring true zero-order-hold output (i.e., output of successive sample levels not separated by noisy TRACK periods, as in simple sample-holds).

(b) Improved Track-hold Circuits. In Fig. 5-38*a*, the same FET-input high-impedance follower (Sec. 3-30) serves both as the capacitor driver in TRACK and as the output amplifier in HOLD, at the expense of extra switching.

In Fig. 5-38*b*, degenerative feedback around the entire sample-hold circuit effectively lowers the input-amplifier output impedance, and thus the capacitor-charging time constant (see also Fig. 5-16).

(c) Switched Integrators as Track-hold Circuits. An operational-amplifier integrator without integrand input, shown in solid lines in Fig. 5-38*c*, will track its RESET input with a charging time constant *rC* and will then hold in its COMPUTE/HOLD mode; the low-impedance output can be loaded without discharging the holding capacitor. The speedup capacitor shown in dashed lines will improve phase shift in TRACK; if we also want to speed up the charging transient on switching into TRACK, we must add the switch *SW*2.

In many general-purpose hybrid computers, integrators serve as track-hold circuits, although it may be a pity to tie up accurate summing

Table 5-3. Typical Specifications of ±10-volt Sample-hold Circuits Used for Analog-to-digital Conversion

---

Gain accuracy: 0.01 to 0.05 percent of half-scale ±0.002 percent of half-scale/deg C
Bandwidth in TRACK *mode:* To several megahertz, full signal, depending on applications; small-signal dynamic error should match gain accuracy at working frequencies; frequently not properly specified (see Sec. 3-2)
Acquisition time (step-response or settling time in TRACK mode): 2 to 10 μsec within gain accuracy
Aperture time (switching time into HOLD): 50 to 500 nsec
HOLD *drift:* 0.1 to 2 mV during typical HOLD period
Noise: 1 mV peak-to-peak (output)
Input-Output crosstalk: 0.01 to 0.04 percent of half-scale

---

integrators in this manner. In the integrator circuit of Fig. 5-34, the TRACK charging time constant is $C_{\text{INT}}\, r_F$ if $C_{\text{HOLD}} = C_{\text{INT}}$, where r_F is the forward resistance of the bipolar-transistor shunt switch. With $r_F = 10$ ohms and $C_{\text{INT}} = C_H = 0.001\ \mu$F, the phase-shift error $200\pi f\, C_{\text{INT}}\, r_F$ percent (Sec. 3-2) in the HOLD output will result unless we add some sort of equalization such as the speedup capacitor in Fig. 5-38*c*; but accurate equalization is difficult, since r_F is a nonlinear resistance.

The charging time constant of integrator-type track-hold circuits like that in Fig. 5-38*c* is dramatically reduced if the switch *SW*1 is driven by a current-amplifying unity-gain follower with low output impedance Z_S. The effective charging time constant is then of the order of $(Z_S + R_{\text{ON}})C$, where R_{ON} is the switch resistance (see also Sec. 5-18).[35,37] In the integrator/track-hold circuit of Fig. 5-35, the switch is enclosed in the follower-amplifier feedback loop (Fig. 5-35*d*), so that the effective charging time constant is only $Z_S C$, where Z_S is of the order of 0.1 ohm at 100 Hz; phase shift in TRACK is further improved by the equalizing networks shown in Fig. 5-35*a*.

Capacitance-driving switching circuits like that in Fig. 5-35 may have to supply substantial currents when following fast signals (up to 30 mA in Fig. 5-35). The resulting storage time of saturated bipolar-transistor switches would lengthen the critical aperture time needed to turn the switch OFF (into HOLD or COMPUTE). For this reason, the switched-amplifier output stage in Fig. 5-35*d* is a *nonsaturating* switch (emitter follower switched OFF by low-current saturating shunt switches), which can switch into COMPUTE/HOLD within 40 nsec. The FET-input switched amplifier of Fig. 5-35*d* was designed as a low-drift replacement for earlier simple switched emitter followers.[32,33,42,43,45]

5-18. Switched Operational Amplifiers (see also Sec. 5-5). Substitution of a resistance for the integrator capacitance *C* in either Fig. 5-34 or 5-35*a* produces an electronic two-input SPDT switch with sign-inverted low-impedance output (switched phase inverter or summer, Fig. 5-39*a*). Figure 5-39*b* shows a more general type of switched operational amplifier implementing

$$X_0 = -\frac{Z_0(P)}{Z_1(P)}X_1 \qquad \text{or} \qquad X_0 = -\frac{Z_0'(P)}{Z_1'(P)}X_1' \qquad \left(P \equiv \frac{d}{d\tau}\right) \tag{5-10}$$

In addition to switching-spike and switch-leakage errors, switched operational amplifiers will have transfer-function errors due to finite amplifier gain and follower and/or switch impedances. Such errors are readily calculated by the methods of Secs. 3-11 to 3-14.[34] The circuit of Fig. 5-39*b*, for example, is an ordinary operational amplifier with the switch OFF; with the switch ON, $X_0 = -(Z_0/Z_1)X_1 + e$. The error $e = e(\tau)$ is given by

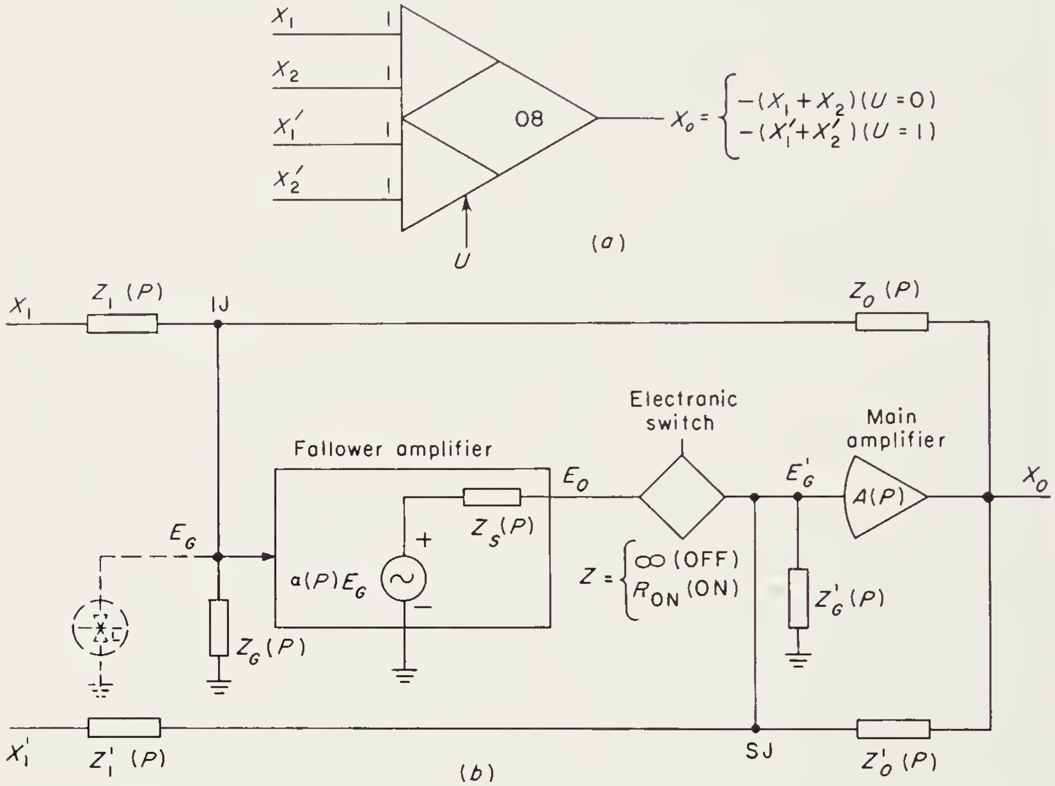


Fig. 5-39. Suggested block-diagram symbol for a switched phase inverter/summer (*a*), and a more general switched operational amplifier (*b*).

$$\left.\begin{aligned} e &\approx -\left(\frac{1}{A_{ON}\beta_{ON}}\frac{Z_0}{Z_1}X_1 + \frac{\beta_1}{\alpha\beta_{ON}}\frac{Z_0}{Z_1'}X_0'\right) \\ A_{ON} &= \frac{\alpha A}{1 - A\beta_1} \qquad \beta_{ON} = \left(1 + \frac{Z_0}{Z_1} + \frac{Z_0}{Z_G}\right)^{-1} \qquad \beta_1 \approx \frac{Z_S + R_{ON}}{Z_0'} \end{aligned}\right\} \tag{5-11}$$

at frequencies where $|A_{ON}\beta_{ON}| \gg 1$ and $|Z_S + R_{ON}| \ll |Z_0|, |Z_1'|, |Z_G'|$.[34] Note that the second error term makes this one-switch circuit less practical than multiswitch circuits if low computing impedances are to be used. Stability considerations usually will not permit follower-amplifier gains α much above unity.

If the switch in Fig. 5-39*b* is enclosed in the follower-amplifier feedback loop in the manner of Fig. 5-35, then $Z_S + R_{ON}$ in Eq. (11) is simply replaced by Z_S.

REFERENCES AND BIBLIOGRAPHY

References 1 to 5 are primary references on device models and circuits for electronic switches. See also the current application bulletins published by manufacturers of semiconductors and of discrete and integrated-circuit operational amplifiers.

General References

1. Millman, J., and H. Taub: *Pulse, Digital, and Switching Waveforms*, McGraw-Hill, New York, 1965.
2. Gray, P. E., et al.: *Physical Electronics and Circuit Models of Transistors*, Wiley, New York, 1965.
3. Phillips, A. B.: *Transistor Engineering*, McGraw-Hill, New York, 1962.
4. Hoeschele, D. F.: *Analog-to-digital/Digital-to-analog Conversion Techniques*, Wiley, New York, 1968.
5. Schmid, H.: *Handbook of Analog/Digital/Analog Conversion*, Van Nostrand Reinhold, New York, 1971.
6. Smith, B. R.: Choppers: A Survey of Low-level Switching, *The Electronic Engineer*, March 1968.
7. Kvamme, E. F., and D. L. Wollensen: Analog Switching, *EDN*, Jan. 15, 1970.
8. Morrill, C. D., and R. V. Baum: A Stabilized Electronic Multiplier, *IRETEC*, December 1952.

Diode Switches

9. Millman, J., and T. H. Puckett: Accurate Linear Bidirectional Diode Gates, *Proc. IRE*, January 1955.
10. Gray, H. J., M. Rubinoff, and J. Tompkins: A Diode Multiplexer for Analog Voltages, *IRETEC*, June 1955.
11. Sebestyen, G.: A Design Technique for Pedestal-free Switching Circuits, *IRETEC*, September 1957.

Bipolar-transistor Switches

12. Bright, R. L.: Junction Transistors Used as Switches, *Communications and Electronics*, **111** (1955).
13. ——— and A. P. Kruper: Transistor Chopper for Stable D-c Amplifiers, *Electronics*, April 1955.
14. Meyer-Brötz, G.: Eigenschaften und Anwendungen von Flächentransistoren als Schalter, *Telefunken Z.*, June 1960.
15. Schneider, W.: Der Transistor als genauer elektronischer Schalter, *Nachr. Tech. Fachberichte*, April 1960.
16. Mitchell, B., and B. Bell: The INCH (Integrated Chopper), *Gen. Eng. Memo* 7, National Semiconductor Corp., Danbury, Conn., June 1962.
17. Transformerless Chopper Drive Circuit, *Gen. Eng. Memo* 8, National Semiconductor Corp., Danbury, Conn., July 1962.
18. Hochwald, W., and H. L. Ehlers: Design of a High-quality Transistor D-c Amplifier, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
19. Schmid, H.: Four-quadrant, All-electronic Pulse-time Multiplier, *Rept.* 62APJ43, General Electric Co., Johnson City, N.Y., 1962.
20. Gulick, J. F.: A High-speed Electronic Switch for Bipolar Analog Signals, *Johns Hopkins Univ., Appl. Phys. Lab. Rept.* CF-3044 (AD-651008), Silver Spring, Md., 1963.
21. Lyden, A.: Single and Matched-pair Transistor Choppers, *Electron. Eng.*, March 1965.
22. Finlay, D. J.: The Analysis and Performance of Transistor Choppers, *Electron. Eng.*, September 1966.

FET Switches

(See also Refs. 4 to 7)

23. Field-effect Low-level Choppers, *File* 103, Siliconix, Inc., November 1963.
24. Barton, K.: The FET Used as a Low-level Chopper, *Electron. Eng.*, February 1965.

25. Bergersen, T. B.: FETs in Chopper and Analog Switching Circuits, *Application Note* AN-220, Motorola Semiconductor, Phoenix, Ariz., 1967.
26. Evans, A. D.: IC's End the Driver Gap, *Electron. Design*, Oct. 25, 1966.
27. Hardcastle, I.: D/A Converters Profit from FET's, *EDN*, June 1, 1969.
28. Mrazek, D.: MOS Commutators, *Electronics*, June 8, 1970.

Reed Relays

29. Cullin, A.: Taking Bounce out of Reed Relays, *Electronics*, July 6, 1962.
30. Fasal, J.: Magnetic Reed Switches and Relays, *Electron. Design*, July 5, 1963.
31. Counting, Selection, and Logic, *Application Manuals* 401–403, C. P. Clare and Co., Chicago, Ill., 1962.

Miscellaneous Switching and Mode-control Circuits

32. Brubaker, T. A.: A Non-saturating Transistor Switch for Analog and Hybrid Computers, *ACL Memo.* 78, Electrical Engineering Department, University of Arizona, May 1963.
33. Baker, R. H., et al.: The Diamond Circuit, *MIT Tech. Rept.* 300, Lincoln Laboratory, Lexington, Mass., January 1963; see also *Proc. Intern. Solid-state-circuits Conf.*, Philadelphia, 1963.
34. Korn, G. A.: Performance of Operational Amplifiers with Electronic Mode Switching, *IEEETEC*, June 1963.
35. ———: New High-speed Analog and Analog-digital Computing Techniques: The ASTRAC System, *Proc. 3d AICA Conf.*, Opatija, Yugoslavia, 1961, Presses Académiques Européennes, Brussels, 1962.
36. ———: ASTRAC Offers New Computing Methods, *Electron. Inds.*, July 1962.
37. Brubaker, T. A.: ASTRAC I Design Performance and Accuracy Studies, Ph.D. Thesis, University of Arizona, May 1962; *Ann. AICA*, April 1964.
38. Eckes, H. R.: Digital Expansion System for ASTRAC I, M.S. Thesis, *ACL Memo* 75, University of Arizona, May 1963; *Ann. AICA*, January 1964.
39. ——— and G. A. Korn: Digital Program Control for Iterative Differential Analyzers, *ACL Memo* 86, Electrical Engineering Department, University of Arizona, 1963; *Simulation*, February 1964.
40. Korn, G. A.: The ASTRAC II Project, *ACL Memo* 89, Electrical Engineering Department, University of Arizona, 1964; *Proc. 4th AICA Conf.*, Brighton, England, 1964.
41. ———: A Simple First-order Hold Circuit, *Simulation*, February 1964.
42. Deveraux, H. R.: Diamond Circuit Measures Phase Shift, *Electronics*, Sept. 21, 1964.
43. Eckes, H. R.: A Fast Mode Control Switch, *IEEETEC*, February 1965.
44. Ash, J. A., and Y. J. Kingma: MOSFET Mode-control Switching, *Simulation*, August 1965.
45. Conant, B. K.: A New Solid-state Iterative Differential Analyzer, *Proc. FJCC*, 1968.
46. Roberge, J. K.: Gated Amplifiers for Data Processing in Space, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Mass., 1966.
47. Kley, A., et al.: Elektronische Schalter für die Integrierersteuerung, *Telefunken Z.*, January 1966.
48. Widlar, R. J.: IC Op Amp Beats FETs on Input Current, *EEE*, December 1969; *Application Note* AN-29, National Semiconductor Corporation, Santa Clara, Calif.
49. Keay, C. S., and J. A. Kennewell: D/A Converter Switches Inputs with TTL Gates, *Electronics*, Dec. 8, 1969.

Digital-to-analog Converters and Ladder-network Design

(See also Refs. 4 and 5)

50. Pearman, C. R., and A. E. Popodi: How to Design High-speed D/A Converters, *Electronics*, Feb. 21, 1964.
51. Pracht, C. P.: A New Digital-attenuator System, *Simulation*, April 1967.
52. Aaron, M. R., and S. K. Mitra: Synthesis of Resistive D/A Conversion Ladders for Arbitrary Codes, *IEEETEC*, June 1967.
53. ———: A Note on the Design of D/A Converters, *IEEETEC*, October 1967.
54. Freeman, J.: Ladder Networks Are Easy to Design, *Electron. Design*, July 5, 1967.

55. Walker, M.: Exploit Ladder-network Design Potential, *EDN*, Feb. 1, 1969.
56. ———: Expand Basic Ladders into Complex Networks, *EDN*, Mar. 1, 1969.
57. Raamot, J.: 18-bit Digital-to-analog Conversion, *Proc. SJCC*, 1970.

Current-switching DACs

58. Stifle, J.: A Current-mode D/A Converter, *Univ. Ill. Coord. Sci. Lab. Rept* R-263, 1965.
59. Schmid, H.: Current Dividers Convert Digital Signals, *Electronics*, Nov. 14, 1966.
60. Rudin, M. B., et al.: A Family of Integrated Circuits for Data Systems, *Proc. FJCC*, 1967.
61. Blatt, H.: Conic Display Generator Using Multiplying D/A Decoders, *Proc. FJCC*, 1967.
62. Rudin, M. B.: New IC Sparks Low-cost A/D-D/A Conversion, *EDN*, May 1968.
63. Palazzini, N.: Comparison of Amplifiers used with Current-summing D/A Converter, *Electron. Capabilities*, Winter 1968.
64. Marjanovic, S., and D. R. Noaks: A High-speed, High-accuracy D/A Converter, *Univ. Birmingham Memo*, Birmingham, England, 1969.

Serial Decoding

65. Centner, R. M., and J. R. Wilkinson: New Approach to Serial Decoding, *Electronics*, Aug. 24, 1962.

Analog-to-digital Conversion

(See also Refs. 4 and 5)

66. Kay, B. G.: Selecting the Right Digital Voltmeter, *Electronics*, Apr. 4, 1966.
67. Schmid, H.: Digital Meters for Under $100, *Electronics*, *Nov.* 28, 1966.
68. Giloi, W., and H. Sommer: PHENO: A New Concept of Hybrid Computing Elements, *Proc. FJCC*, 1967.
69. Crawford, J. I., and M. J. Bodoia: The IADIC: A Hybrid Computing Element, *Proc. FJCC*, 1967.
70. Aasnaes, H. B., and T. J. Harrison: Triple Play Speeds A/D Conversion, *Electronics*, Apr. 29, 1968.
71. DeLotto, I., et al.: A Fast Amplitude-to-time Converter for an Equivalent Clock Rate of Some GHz, *Nucl. Instr. Methods*, **65**:228 (1968).
72. Manfredi, P. F., et al.: Very Precise Method for Fast Analog-to-digital Conversion, *Alta Frequenza*, November 1968.
73. Schick, L. L.: A Comparatorless Dual-slope Digital Voltmeter, *Trans. IEEE Instr.*, September 1968.
74. Goltz, J. R.: Economical A/C Conversion for Hybrid Computers, *Instr. Control Systems*, October 1968.
75. Savitt, A.: A High-speed Analog-to-digital Converter, *IRETEC*, March 1959.
76. Waldhauer, F. D.: Analog-to-digital Converter, U.S. Patent 3,187,325 (assigned to Bell Telephone Laboratories, Inc.), 1965.
77. 1-Mc Cyclic A/D Converter, *Electron. Products*, March, 1968.

Miscellaneous

78. Marjanovic, S., and D. R. Noaks: Design Criteria for the Mode Control of Hybrid-computer Amplifiers, *Proc. IEE* (Great Britain), April 1970.
79. Korn, G. A.: Bit Switch for a Calibration-free MDAC, *Univ. Arizona CSRL Memo* 201, February 1970.
80. Cohen, J. M.: Sample-and-hold Circuits Using FET Analog Gates, *EEE*, January 1971.

CHAPTER 6

ELECTRONIC MULTIPLIERS, DIVIDERS, AND RESOLVERS

INTRODUCTION

6-1. Survey. Early analog computers implemented multiplication with *computer servomechanisms.*[1–21] A single servo could position between 4 and 20 potentiometers for multiplication and function generation, but servo frequency response is necessarily limited (Fig. 6-1), and maintenance of electromechanical components is a nuisance. High-quality electronic multipliers are not only faster but also more accurate than the best servomultipliers.

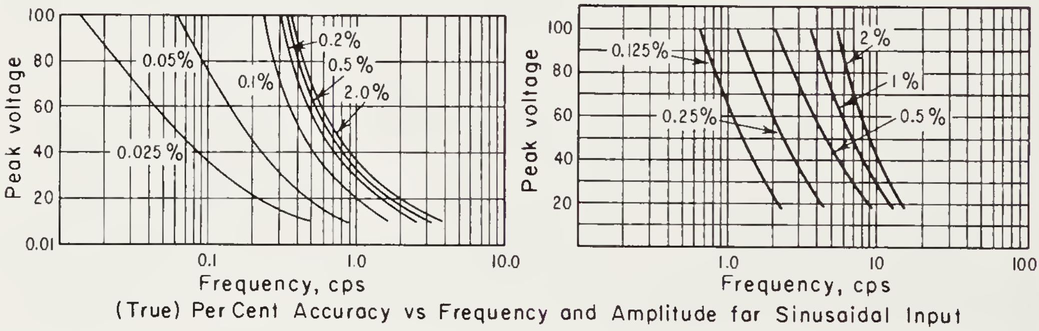


Fig. 6-1. Performance of a high-accuracy 60-Hz servomultiplier employing 10-turn wire-wound potentiometers and of a high-speed 400-Hz servo using single-turn film potentiometers. Each servo can position up to four multiplying potentiometers. (*Electronic Associates, Inc.*)

The most frequently employed types of electronic multipliers are:

1. *Diode quarter-square multipliers,* which implement the relation

$$XY = \tfrac{1}{4}[(X + Y)^2 - (X - Y)^2]$$

by means of diode-function-generator squaring circuits (Sec. 6-2). Such multipliers are almost exclusively used in general-purpose analog/hybrid computers, since errors can be within 0.05 percent of half-scale at d-c and 0.5 percent of half-scale up to 2 and even 10 KHz (Secs. 6-2 and 6-3).

2. *Multipliers employing devices with voltage-controlled gain.* In this category, *transistor-transconductance multipliers* are practically the only ones to survive; they combine moderate static accuracy (0.5 to 2 percent of half-scale) with remarkable bandwidth (up to 1 MHz for 2 percent of half-scale dynamic errors (Secs. 6-5 to 6-8).
3. *Time-division or pulsed-attenuator multipliers* based on simultaneous pulse-amplitude and pulse-width modulation of a pulse train. Such multipliers are potentially more accurate (0.01 percent of half-scale at d-c) as well as more expensive than quarter-square multipliers, but phase-shift and transient errors associated with ripple filtering have restricted time-division multipliers to low-bandwidth applications (Sec. 6-9).

Other multiplier circuits sometimes encountered are based on *triangle-wave averaging* (Sec. 6-4), *analog/digital/analog conversion* (digital servos, Sec. 6-8), *logarithmic function generators* (Sec. 4-16), and *photoresistor gain control* for multiple (as many as 10 to 20) products (Sec. 6-8). A vast array of other electronic multiplication schemes involving variable-gain devices, special modulation systems, and even cathode-ray-tube circuits have been tried and abandoned. A short bibliography on this subject is appended to this chapter, since one or the other old circuit trick might, some day, become useful in connection with new devices or technology.

QUARTER-SQUARE MULTIPLIERS AND RELATED CIRCUITS

6-2. Diode Quarter-square Multipliers. Low-cost multipliers implementing the "quarter-square" relation

$$XY = \tfrac{1}{4}[(X + Y)^2 - (X - Y)^2] \qquad (6\text{-}1)$$

have used square-law transfer characteristics of vacuum tubes,[24,25] temperature-compensated varistor networks,[26–30] and field-effect transistors[31] to produce static accuracies between 0.2 and 0.5 percent of half-scale. Such circuits are obsolete, since transconductance multipliers (Sec. 6-6) are as accurate, cheaper, and faster.

Diode squaring circuits (see also Sec. 4-15) approximate the desired parabolic transfer characteristic by straight-line-segment approximations slightly improved by natural rounding at the breakpoints. Since

$dx^2 = 2x\,dx$, uniform breakpoint spacing conveniently permits us to use identical slope increments and hence identical diode-limiter channels on each side of zero. For uniform breakpoint spacing, polygonal-approximation errors can be kept within ϵ percent of half-scale with $10/\sqrt{\epsilon}$ segments;[33,37] proper adjustment will yield equal positive and negative error peaks for all segments (Fig. 6-2). Nonuniform breakpoint spacing, discussed in Refs. 33 and 37 to 39, will not reduce the required number of diode channels appreciably; it may, however, be worthwhile to add two extra breakpoints near zero in order to minimize zero-input errors.

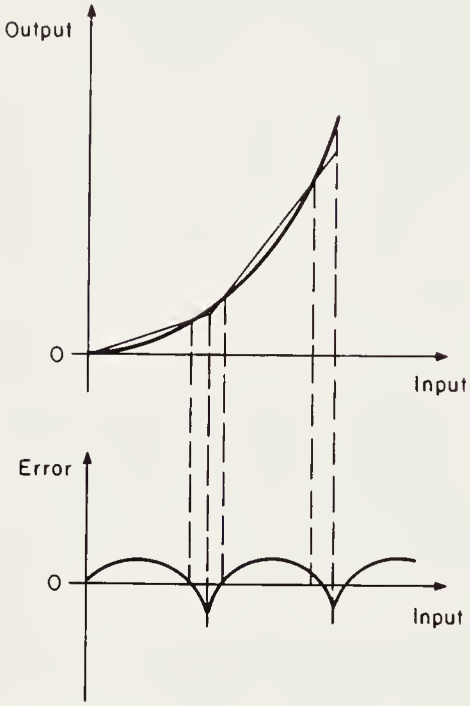
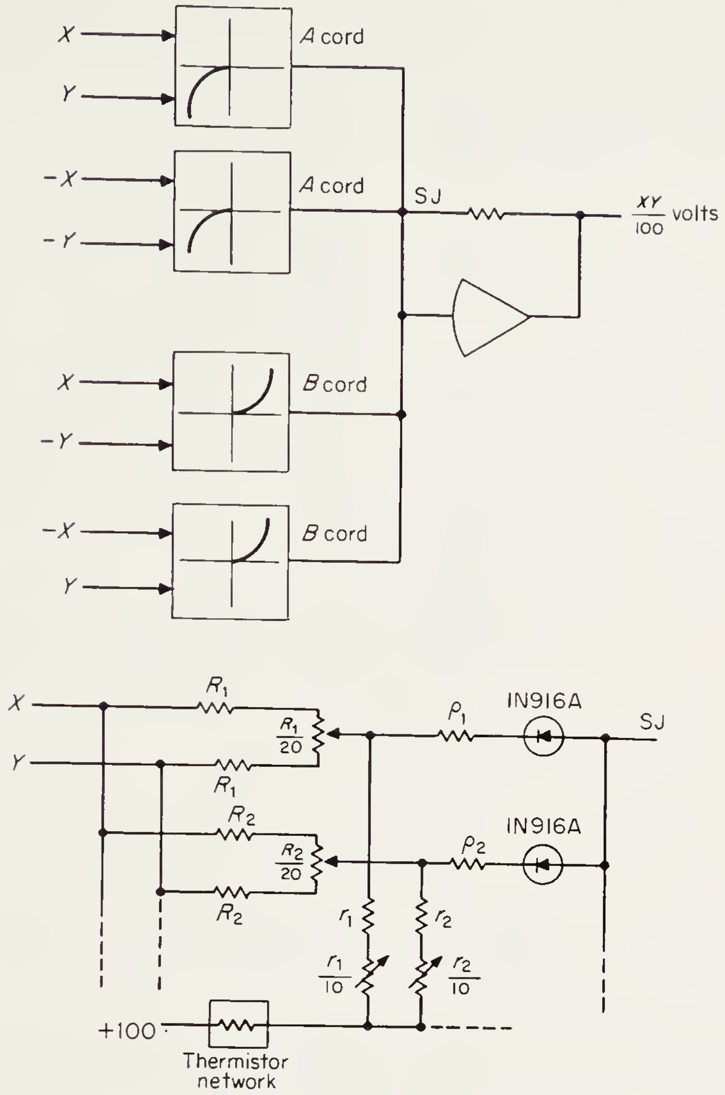


Fig. 6-2. Polygonal approximation of a square-law characteristic. Slopes and breakpoints are set for equal positive and negative errors. Additional breakpoints near zero input could be used to reduce the percentage error for low-input voltages.

Since an increase in the number of diodes for improved static accuracy also tends to accumulate the effects of diode capacitances and breakpoint drift, 10- to 150-KHz triangular or sinusoidal interpolation dither (Sec. 4-15) has been employed in some diode quarter-square multipliers to reduce static errors. Unfortunately, the dither voltage must be removed from the multiplier output by means of a ripple filter, which introduces phase shift and thus deteriorates the multiplication bandwidth.

Figures 6-3 and 6-4 show high-quality quarter-square multiplier circuits. Note that diode drift will not affect the zero error, since no series diodes conduct for $X = Y = 0$. Although input voltages $-X$, $-Y$ as well as X, Y are required because of the nonmonotonic nature of the square function, amplifier connections are simple. While earlier general-purpose computers used patched amplifiers with diode-multiplier cards, modern machines employ committed amplifiers, which are no longer expensive, do not expose the sensitive summing point, and can be nicely equalized for optimal frequency response. The absolute-value squaring circuits in Fig. 6-4 reduce costs and simplify circuit adjustment. The wideband multiplier of Fig. 6-4*a* and *b*[36] employs relatively low circuit impedances and combined series/shunt limiting to minimize diode-capacitance effects (Sec. 4-5).



k	1	2	3	4	5	6	7	8	9	10
R_k	200K	200K	200K	200K	200K	160 K	120 K	80 K	37 K	200 K
r_k	100 K	120 K	140 K	160 K	200 K	200 K	200 K	200 K	200 K	∞
ρ_k	0	0	0	0	0	14 K	30 K	50 K	70 K	100 K

100-volt A cord (reverse diodes and bias for B cord)

Fig. 6-3. A four-card diode multiplier for ±100-volt computers. Static accuracy is within 0.1 percent of 100 volts (0.03-volt error for zero output), and phase shift is within 2 deg up to 1 KHz.

The fixed-function diode networks used in quarter-square multipliers can be temperature-compensated by temperature-sensitive resistors or diodes in the bias circuits (Figs. 6-3 and 6-4; see also Sec. 4-7) and show little drift with aging. References 36 and 41 give detailed procedures for diode temperature compensation. If desired, one can periodically recalibrate each diode squaring card by comparing the output voltage at each breakpoint with the voltage from a precision attenuator furnished with the multiplier package.

The multiplier scheme of Fig. 6-5 implements the modified quarter-square relation

$$\frac{XY}{100} = F_1\left(\frac{Y - X}{2}\right) + F_2\left(-\frac{X + Y}{2}\right) - 3X \tag{6-2}$$

where $$F_1(U) \equiv -3U - \frac{U^2}{100} \qquad F_2(V) \equiv -3V + \frac{V^2}{100} \tag{6-3}$$

are seen to be *monotonic* functions of the card-input voltage.[34,35]

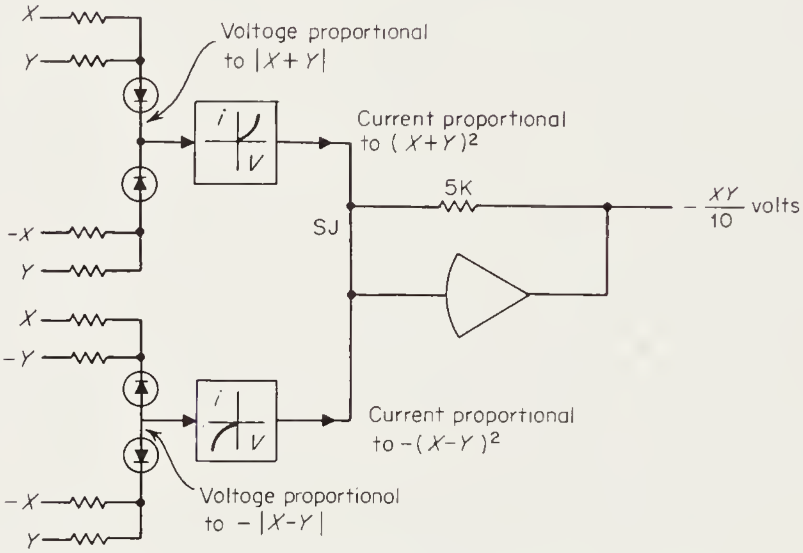

X
Y
-X
Y
X
-Y
-X
Y
Voltage proportional to |X+Y|
i
V
Current proportional to (X+Y)²
5K
SJ
$-\frac{XY}{10}$ volts
i
V
Current proportional to -(X-Y)²
Voltage proportional to -|X-Y|

Fig. 6-4*a*. A fast ±10-volt multiplier using two squaring cards with absolute-value circuits. Static accuracy within 0.2 percent of 10 volts between 15 and 40 deg C. Dynamic errors ($X = 10$, $Y = 10 \sin \omega t$) are within 0.2 percent of 10 volts at 1 KHz and within 0.5 percent up to 10 KHz. (*University of Arizona, Ref.* 36.)

6-3. High-accuracy Squaring Circuits.[42–47] If great bandwidth is not needed, the number of diode segments required for a given polygonal square-law approximation can be reduced with the aid of the relation

$$x^2 = \tfrac{1}{4}(u^2 + 2u + 1) \qquad \text{with } u = 2\,|x| - 1 \tag{6-4}$$

With proper scaling, squaring-circuit errors are reduced by a factor of 4. Suitably chosen n-breakpoint approximations to u^2 (or $n/2$-breakpoint approximations to $|u|^2$) for $-1 \le u \le 1$ will yield $2n$-breakpoint approximations to x^2 ($-1 \le x \le 1$). Note also that x^2 is a monotonic function of u. The relation (4) can be applied recursively to "fold" u^2 in turn; at the last stage, u^2 can be simply replaced by $|u|$.

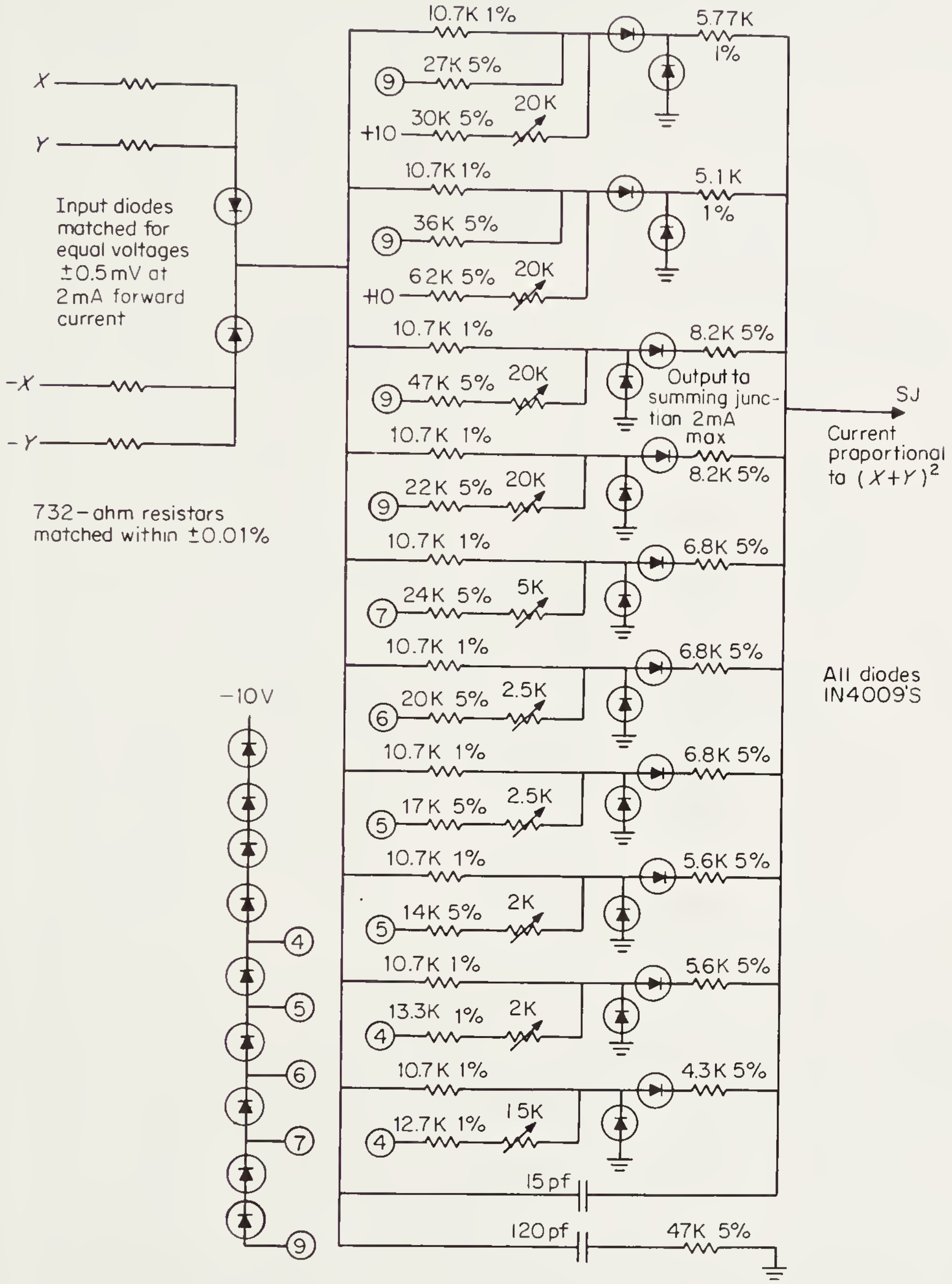
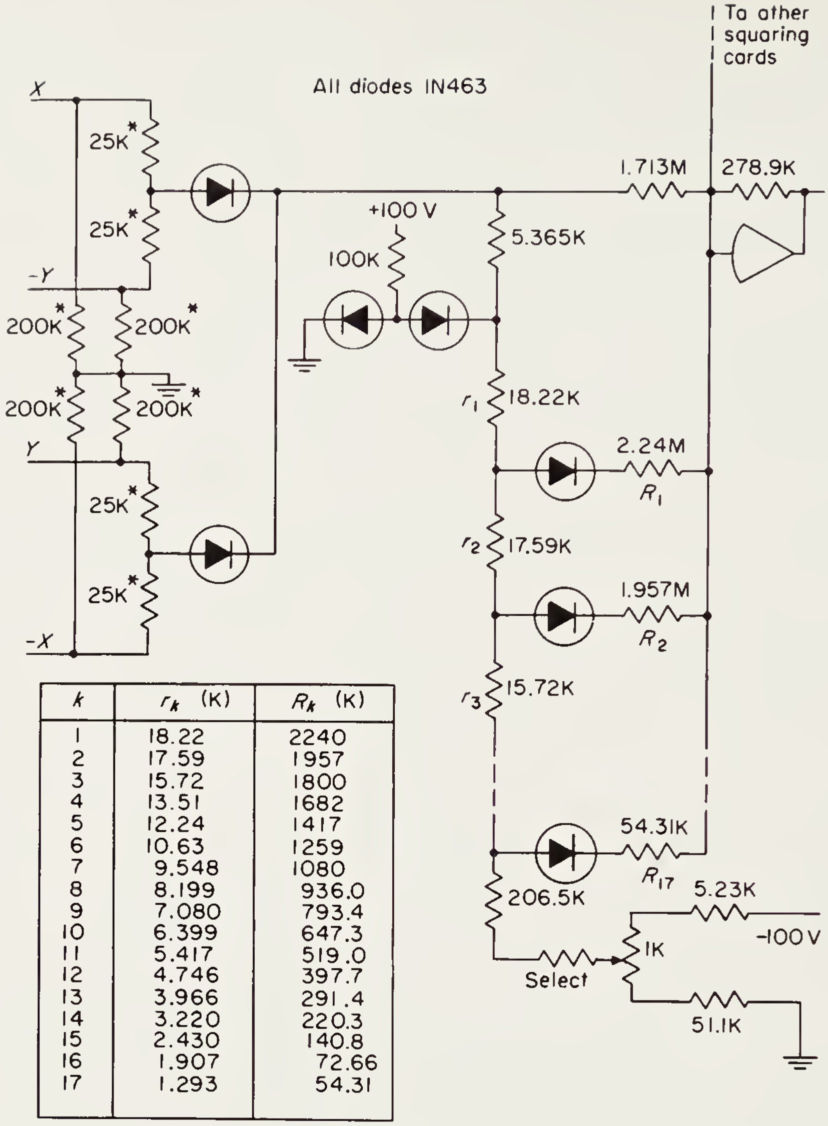


Fig. 6-4*b*. Positive squaring card from Fig. 6-4*a*; reverse diodes and bias for negative squaring card. Note the use of combined series/shunt limiters and compensating capacitors in this wideband low-impedance design, and the temperature-compensating diode string supplying the bias voltages. (*University of Arizona*, *Ref*. 36.)



k	r_k (K)	R_k (K)
1	18.22	2240
2	17.59	1957
3	15.72	1800
4	13.51	1682
5	12.24	1417
6	10.63	1259
7	9.548	1080
8	8.199	936.0
9	7.080	793.4
10	6.399	647.3
11	5.417	519.0
12	4.746	397.7
13	3.966	291.4
14	3.220	220.3
15	2.430	140.8
16	1.907	72.66
17	1.293	54.31

Fig. 6-4*c*. A 100-volt diode network (B card) with absolute-value circuit (reverse diodes and bias for A card). Diodes are selected for low leakage (1 nA at −36 volts); absolute-value diodes are matched within 1 mV at 0.75 mA. 0.5 percent deposited-carbon resistors are used, and starred resistors are matched within 0.01 percent. Multiplier short-term static accuracy is within 0.15 percent of the 100-volt half-scale. (*Systron-Donner Corporation.*)

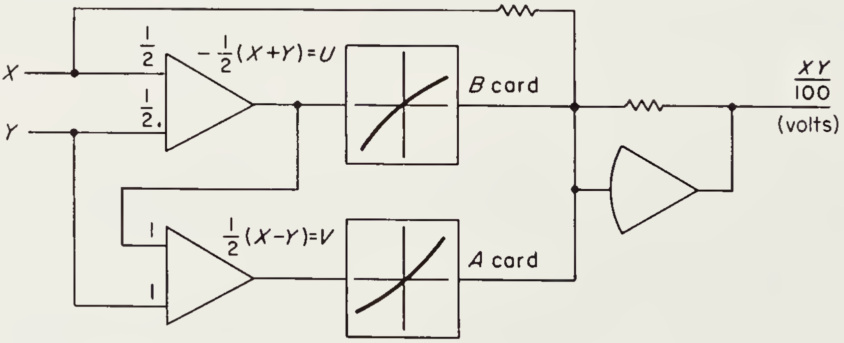


Fig. 6-5. A modified quarter-square multiplier using monotonic diode-function cards.

At frequencies up to about 2 KHz, the absolute-value functions required for such squaring operations can be implemented very accurately with the precision-limiter circuits of Sec. 4-10 (Fig. 6-6).

A single application of Eq. (4) yields the crude but very simple product approximation

$$xy = \left(\frac{x+y}{2}\right)^2 - \left(\frac{x-y}{2}\right)^2$$
$$\approx \tfrac{1}{2}(|x+y| - |x-y|) + \tfrac{1}{4}\big(\big||x+y| - 1\big| - \big||x-y| - 1\big|\big) \tag{6-5}$$

6-4. Triangle-wave-averaging Multipliers.[48–52] Triangle-wave-averaging multipliers are quarter-square multipliers based on the fact that a biased and clipped triangular waveform has a d-c component proportional to the square of the bias voltage (Fig. 6-6). In Fig. 6-7*a*, triangular waveforms $bV(t)$ respectively biased by $a(X - Y)$ and $-a(X + Y)$ are clipped at zero to produce the limiter-output voltages $e_1(t)$, $e_2(t)$ shown in Fig. 6-7*b*. The low-pass output-amplifier circuit averages the shaded triangle portions and adds aY to produce the desired output

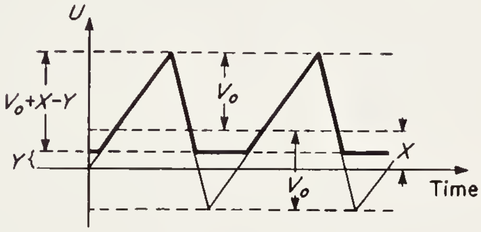


Fig. 6-6. Principle of triangle-averaging multiplication: the running average of the clipped triangular waveform shown in solid lines is $(1/4V_0)(V_0 + X - Y)^2 + Y$.

$$-\frac{\alpha b V_0}{a^2}\left\{\frac{1}{4bV_0}[bV_0 + a(X - Y)]^2 - \frac{1}{4bV_0}[bV_0 + a(X + Y)]^2 + aY\right\} = \alpha XY \tag{6-6}$$

Note that the triangle amplitude V_0 must be accurately constant. As in the case of modulation-type multipliers (Sec. 6-8*d*), the bandwidth of a triangle-averaging multiplier is essentially that of the averaging filter; since lower carrier frequencies permit more accurate clipping, each design requires a compromise between static and dynamic accuracy, as shown in Sec. 6-12. Many different triangle-averaging multipliers have been designed. Relatively good static accuracy (0.1 percent of half-scale) can be obtained at the expense of bandwidth (1 percent of half-scale dynamic error at 1 KHz) through the use of precision limiters (Sec. 4-10) at low (5 to 10 KHz) carrier frequencies.[51,52] To cancel errors due to imperfect clipping and triangle generation at higher carrier frequencies, one may average more than two clipped triangular

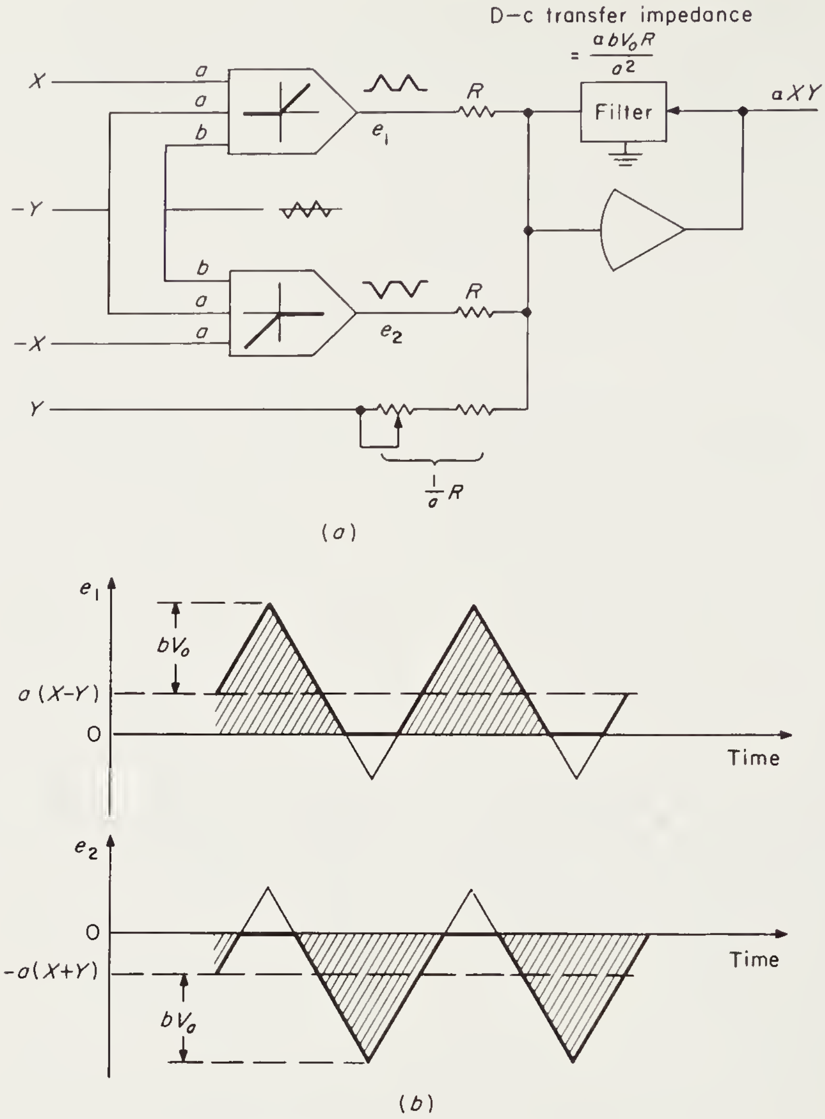


Fig. 6-7. The basic triangle-averaging multiplier (*a*) produces limiter-output waveforms (*b*) whose running averages are added together with aY to produce the desired output

$$-\frac{\alpha b V_0}{a^2}\left\{\frac{1}{4bV_0}[bV_0 + a(X - Y)]^2 - \frac{1}{4bV_0}[bV_0 + a(X + Y)]^2 + aY\right\} = \alpha XY$$

waveforms to obtain a sort of push-pull operation.[50] Several triangle-integration multipliers can share a single triangle generator; but accurate four-quadrant multipliers of this type tend to use more amplifiers and may also be harder to adjust than time-division multipliers of comparable performance. As a result, the most interesting triangle-averaging multipliers are, perhaps, those designed for low static accuracy (0.5 to 1 percent of half-scale) with 10-KHz to 2-MHz carrier frequencies at very low cost.[50,65]

Triangular waveforms are readily obtained through integration of square waves from astable multivibrators and from operational-amplifier oscillators of the type discussed in Sec. 4-12.[51]

MULTIPLICATION THROUGH VOLTAGE-CONTROLLED GAIN

6-5. Multiplication of Positive and Negative Inputs. Many electronic multipliers may be represented by the simple block diagram of Fig. 6-8. The voltage gain $K = X_0/X$ of such a device is variable and increases or decreases monotonically with a second input voltage Y. The output voltage will be proportional to the desired product XY if the gain can be made at least approximately proportional to the input Y (linear gain control).

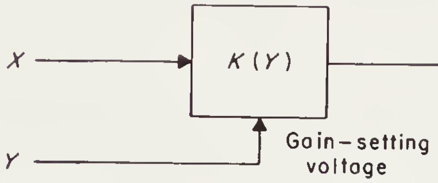


Fig. 6-8. A voltage-controlled variable-gain device.

The ideal electrical multiplier would accept both positive and negative input voltages X, Y, and its output voltage X_0 would have the correct sign associated with the product XY. Such a multiplier is said to be capable of *four-quadrant operation.* Some practical multiplying devices permit only *two-quadrant operation:* one of the two inputs, say X, must remain either positive or negative for proper operation, while Y can change sign and produces the correct output-voltage polarity. A third class of multipliers can accept only voltages of one sign for both X and Y (*one-quadrant multipliers*).

Two-quadrant multipliers can readily produce four-quadrant multiplication by one of the biasing schemes of Fig. 6-9. In Fig. 6-9*a*, a one-quadrant multiplier multiplies Y by $\frac{1}{2}(X + E)$, where E is a positive voltage sufficiently large to keep the input voltage $\frac{1}{2}(X + E)$ positive even when X is negative. The multiplier produces the product $-(\alpha/2)(XY + EY)$, and the output summing amplifier subtracts the undesired term $-(\alpha/2)(EY)$ and yields the product αXY with the correct sign in each case.

The circuit of Fig. 6-9*b* subtracts the outputs of two two-quadrant multipliers to yield the correct product and may cancel some errors because of the symmetrical push-pull arrangement. The biasing

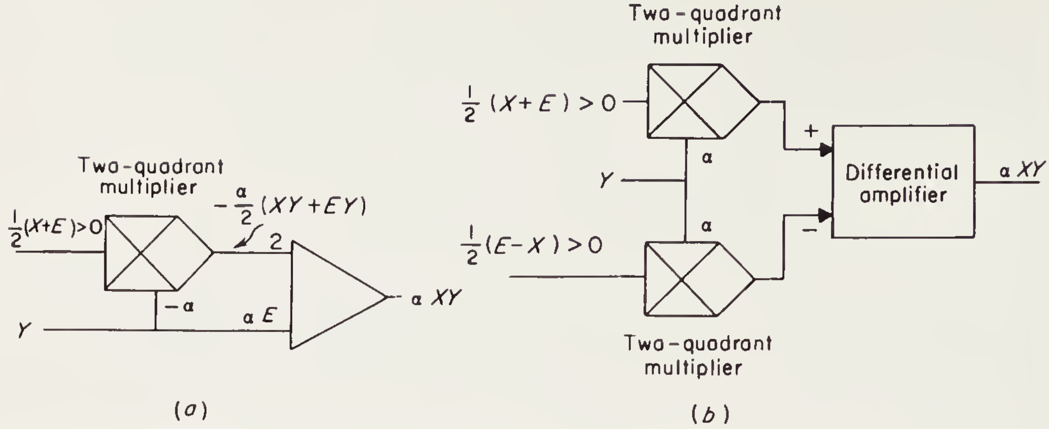


Fig. 6-9. Adaptation of two-quadrant multipliers to four-quadrant multiplication. Analogous schemes permit four-quadrant multiplication with one-quadrant multipliers.

schemes of Fig. 6-9*a* and/or *b* can also be applied to both inputs of a one-quadrant multiplier to yield four-quadrant multiplication.

6-6. Transistor Transconductance Multipliers.[53–57] *Transistor transconductance multipliers* are based on the fact that the gain (transconductance) of a transistor amplifier increases with the transistor emitter current. Elegant differential circuits with matched transistors can cancel undesired bias terms in the output, improve linearity, and permit four-quadrant multiplication. Thus, the cross-coupled differential amplifier circuit of Fig. 6-10*a* uses matched transistors to produce an output voltage X_0 approximately given by

$$X_0 = \alpha(V_1 - V_2)(i_1 - i_2) \tag{6-7}$$

Detailed error analysis on the basis of semiconductor models[54,57] shows that the accuracy of the idealized performance equation (7) depends on accurate matching of the four transistors in Fig. 6-10*a* for V_{BE} vs. collector current and temperature. The output is then quite accurately linear with respect to the currents i_1 and i_2, which may be derived from corresponding voltage inputs with the aid of the differential current source shown in Fig. 6-10*b*. To improve the linearity with respect to V_1 and V_2, the *linearizing differential amplifier* of Fig. 6-10*c* employs the base-emitter junctions of two more matched transistors as diodes in its load circuit. The complete circuit of Fig. 6-10*a*, *b*, and *c* produces the output voltage

$$X_0 = 0.1(X_1 - X_2)(Y_1 - Y_2) \qquad \text{volts}$$

where X_1, X_2, Y_1, Y_2 are input *voltages*.

While the multitransistor combination of Fig. 6-10*a*, *b*, and *c* may look like a fairly complex circuit, it lends itself particularly well to monolithic integrated-circuit construction (one chip for the multiplier

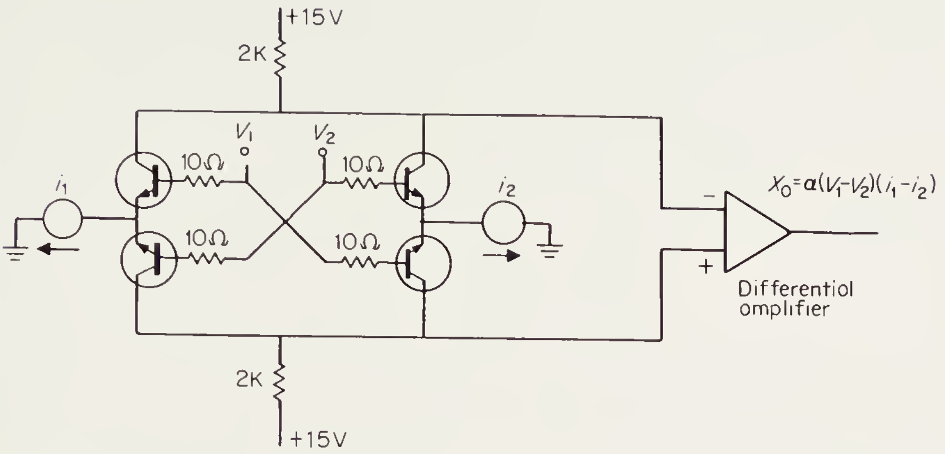


Fig. 6-10*a*. A transistor-transconductance multiplier circuit with differential inputs. Each transistor acts as an amplifier whose gain is controlled by the emitter current; the differential circuit cancels d-c terms and improves linearity. High-beta transistors similar to type 2N2222 are suitable.

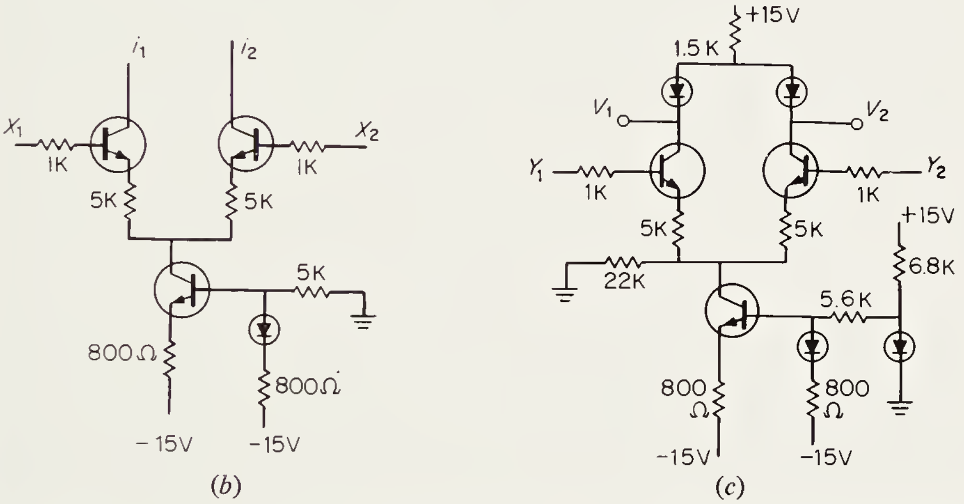


Fig. 6-10*b*, *c*. Differential-amplifier current source for the multiplier circuit of Fig. 6-10*a* (*b*), and linearizing differential amplifier for supplying the voltages V_1, V_2 (*c*). Diodes are transistor base-emitter junctions matching other transistors in the circuit.

circuits proper, plus one for the output amplifier).[57] Depending on quantity and accuracy, such multiplier circuits sell for between $10 and $70.

With careful transistor matching (monolithic construction) and adjustment, multipliers employing the circuits of Fig. 6-10[57] can keep static errors within 0.2 percent of half-scale for ±10-volt inputs and output. Temperature drift can be within ±0.05 percent/deg C for gain and ±1 mV/deg C for offset between −30 and +85 deg C, and dynamic errors reach 1 percent at about 10 KHz. If one is satisfied with 2 to

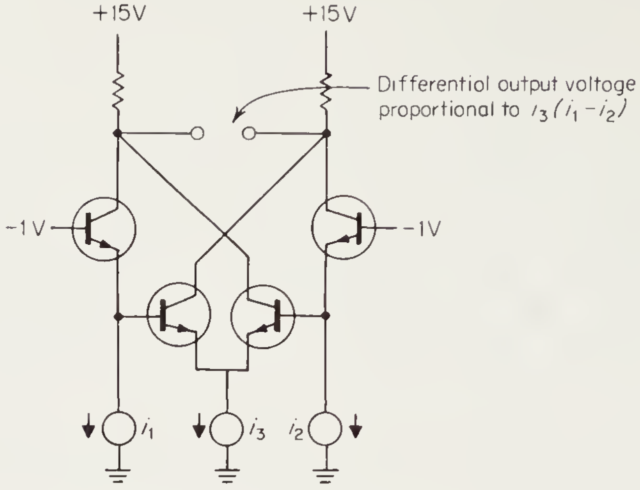


Fig. 6-11. Another transistor-transconductance multiplier circuit, suitable for wideband operation. (*Ref.* 55.)

5 percent accuracy, similar multipliers work nicely up to *several hundred megahertz*. Monolithic construction makes them especially attractive as balanced modulators in communication circuits. Figure 6-11 shows a slightly different transconductance-multiplier circuit suitable for wideband operation.[55,56]

Haase[54] supplied the currents i_1, i_2 in Fig. 6-10*a* from two operational-amplifier current sources (Fig. 1-11) and linearized the multiplier gain with respect to $V_1 - V_2$ with the aid of a feedback loop and "slave" multiplier sections in the manner of Sec. 6-7. Because of the difficulty of matching control and slave sections accurately (Sec. 6-7), static accuracy was about the same as that of the multiplier of Fig. 6-10, with somewhat reduced bandwidth.

6-7. Feedback Control of Multiplier Gain. (a) The Basic Feedback Scheme. The output voltage $XK(Y)$ of a voltage-controlled variable-gain device or modulator/demodulator is not usually a linear function of the gain-setting voltage Y over a wide range of inputs Y, as required for multiplication. Frequently, however, the multiplier gain can be made very accurately proportional to Y with the aid of a *gain-setting feedback*

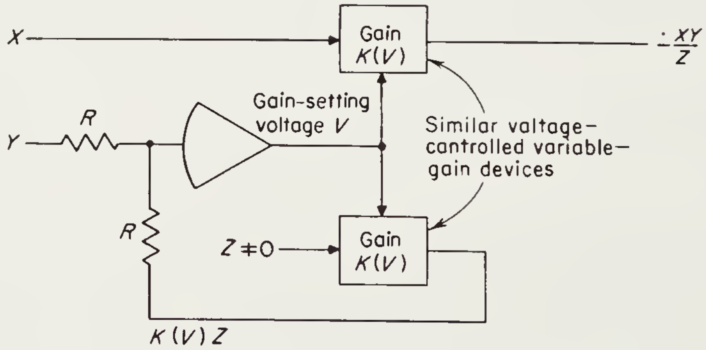


Fig. 6-12. Feedback scheme for making the gain of two or more similar voltage-controlled-gain devices accurately proportional to a multiplier input voltage Y.

loop. Referring to Fig. 6-12, Z is a constant positive or negative reference voltage (e.g., +100 volts or −100 volts) applied to a variable-gain device whose gain K is assumed to be independent of Z. $K = K(V)$ varies monotonically (but not necessarily linearly) with the gain-setting voltage

$$V = A(KZ + Y)$$

supplied by a summing network and d-c amplifier of combined high gain A. It follows that

$$K = K(V) = K(0) + K_1(V)V$$

where K_1 is either positive or negative. Hence

$$K = K(0) + AK_1(V)(KZ + Y) = \frac{K(0) + AK_1(V)Y}{1 - AK_1(V)Z}$$

$$= -\frac{Y}{Z} - \frac{1}{AK_1(V)Z}\left[K(0) + \frac{Y}{Z}\right] \pm \cdots \tag{6-8}$$

if the resulting feedback is stable and has a large loop gain $|AK_1(V)Z|$. A necessary (but not sufficient) condition for stability is $AK_1(V)Z < 0$ at d-c. Additional terms in Eq. (8) are of higher order in $1/AK_1(V)Z$. It is seen that *the feedback regulates the gain K so that it is proportional to the multiplier input Y,*

$$K = -\frac{Y}{Z} \tag{6-9}$$

with an error inversely proportional to the absolute loop gain $|AK_1(V)Z|$.

Assume now that it is possible to obtain a second variable-gain device essentially similar to the first one, i.e., with the same gain/voltage characteristic $K(V)$. If both variable-gain blocks are controlled by the amplifier output voltage V in Fig. 6-12, the second variable-gain device will multiply its input voltage X by $K = -Y/Z$ to yield the desired multiplier output $-XY/Z$. If additional variable-gain devices with identical gain/voltage characteristics $K(V)$ are controlled by the gain-control loop of Fig. 6-12, they can multiply additional inputs X', X'', . . . by $K = -Y/Z$ (*slave multipliers*).

(b) Design Problems. The multiplication scheme of Fig. 6-12 will require the design of a stable high-gain feedback loop to implement Eq. (8) accurately. This is usually not difficult, but every implementation of Fig. 6-12 poses two serious problems. One requires:

1. *A type of voltage-controlled variable-gain device whose gain is accurately independent of one input* (Z or X in Fig. 6-12)
2. *Two such devices with accurately similar gain vs. gain-setting voltage characteristics* $K(V)$

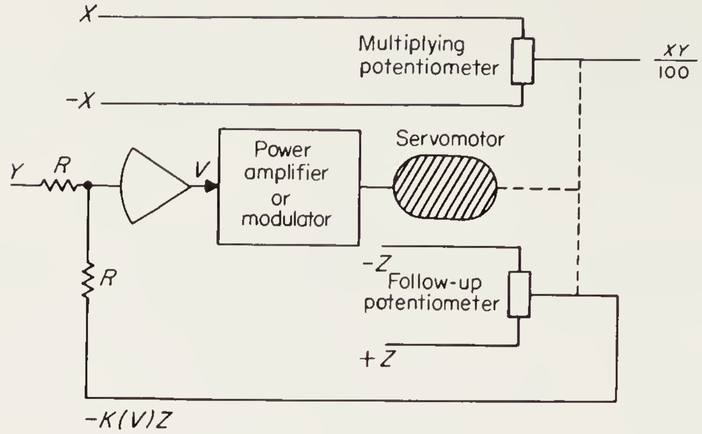


Fig. 6-13*a*. Servomultiplier.

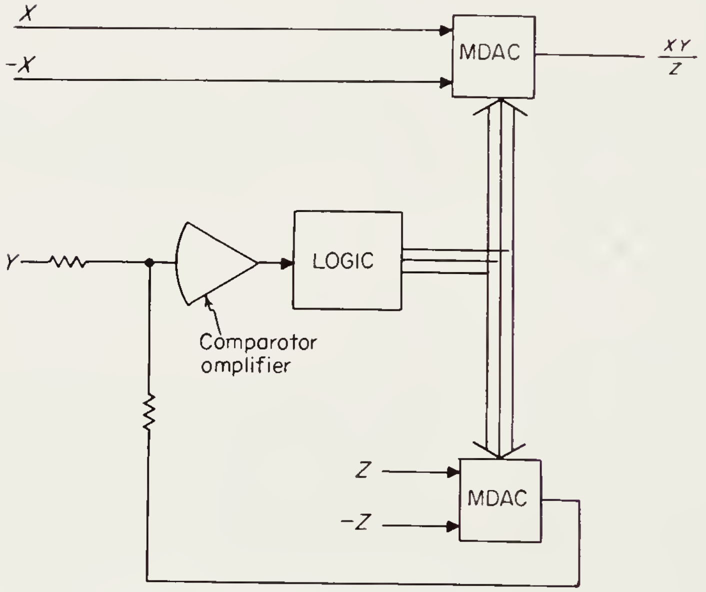
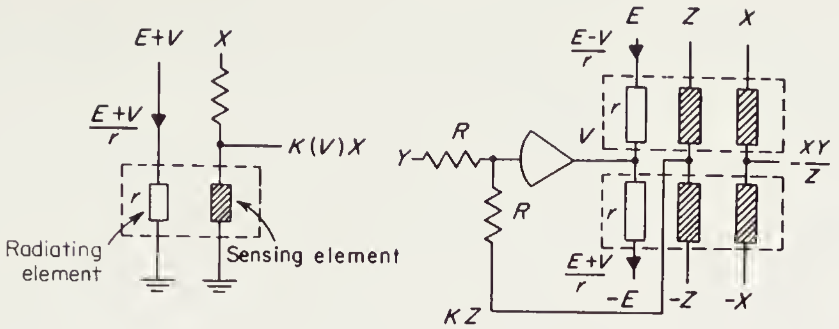


Fig. 6-13*b*. A hybrid analog-digital multiplier.

Practically all simple voltage-sensitive circuit elements, such as Thyrite resistors or semiconductors, fail to satisfy either of these two requirements; their gain may be dependent on the input voltage, and their characteristics cannot be reproduced with sufficient accuracy. One can overcome these deficiencies (1) by using one and the same variable-gain device to attenuate both Z and X on a time-sharing or frequency-sharing basis[143] or, again, (2) by employing semiconductors only as *switches* which are either off or on (saturated). Their specific nonlinear characteristics will then produce only relatively small effects on the circuit output.



(c) (d)

Fig. 6-13*c*, *d*. Principle of photoresistor and heat-transfer multipliers. Radiating elements could be No. 338 incandescent lamps or No. 40 nichrome heating filaments. Sensing elements are commercially available photoresistors (e.g. Clairex CL-605L-020) or heat-sensitive resistors (e.g., No. 40 Teflon insulated nickel wires wound with the heater wire on a Pyrex rod). Circuit and construction details are given in Refs. 60 to 63.

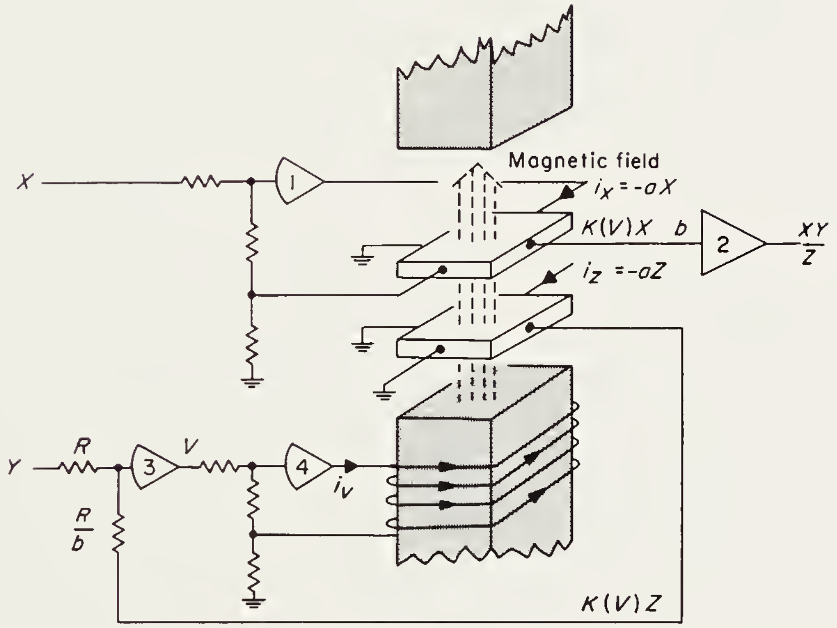


Fig. 6-13*e*. Feedback Hall effect multiplier using two similar Hall effect transducers in the air gap of an electromagnet. Amplifier 1 has current feedback (Sec. 1-17) to produce an output current proportional to the multiplier input voltage X. Amplifiers 3 and 4 form a single amplifier having a current-feedback output stage preceded by voltage amplification. The reference current i_Z is obtained from a high-impedance voltage source whose sign is chosen so as to keep the gain-setting feedback loop stable.

6-8. Examples of Feedback Multipliers. (a) Servo- and Analog-digital Multipliers. The best known example of a feedback multiplier is the electromechanical *servomultiplier* (Fig. 6-13*a*). The variable-gain devices are *potentiometers driven by a motor*, which is, in turn, controlled by the output V_0 of the servoamplifier. $K = -Y/Z$ can be positive or negative. Static accuracy may be within 0.05 percent of half-scale.

Up to at least 20 "slave" products can be obtained with additional potentiometers, but bandwidth is limited by the electromechanical components (see also Fig. 6-1).

In the *hybrid analog-digital multiplier* of Fig. 6-13*b*, the variable-gain devices are *multiplying digital-to-analog converters* (*MDACs*, Sec. 5-6) *controlled by a feedback-type analog-to-digital converter* ("continuous" or successive-approximation converter, Sec. 5-13*b*) as the gain-control loop.[64–66] Such multipliers may have any number of slave sections and can be more accurate than any other analog multiplier at d-c (within 0.005 percent of half-scale). Their bandwidth is limited by their sampled-data output; i.e., the data-reconstruction errors derived in Sec. 9-3 must be considered. From Table 9-1, 0.1 percent reconstruction accuracy requires 6,280 samples per sine-wave cycle with zero-order-hold output or 200 samples with first-order-hold output. Conversion rates possible at reasonable cost range between 20,000 and 500,000 samples/sec, depending on accuracy (see also Sec. 5-13), so that analog-digital multipliers can serve to up to a few hundred hertz with zero-order-hold output and up to a few kilohertz with first-order-hold output. Section 6-14 indicates a method for correcting the sampled-data errors with the aid of an added wideband low-accuracy analog multiplier.

(b) Photoresistor and Heat-transfer Multipliers.[60–63] In another class of feedback multipliers, the variable-gain devices are *light- or heat-sensitive resistors with resistances varied by a voltage-controlled incandescent or neon lamp or a heating filament.* In Fig. 6-13*c*, the gain changing voltage V is added to the nominal operating voltage E of the radiating element (light bulb, neon lamp, or heater filament). The resulting light- or heat-output variation changes the resistance of the sensing element (light- or heat-sensitive resistor), and thus the gain K. Figure 6-13*d* shows a feedback-controlled push-pull multiplier permitter four-quadrant operation. Several slave products can be obtained with additional transducers.

Photoresistor and heat-transfer multipliers have limited accuracy (within 1 percent of half-scale) and low bandwidth (usually less than 10 Hz for 1 percent dynamic error). Their chief virtue is low cost for multiple products, but wideband integrated-circuit transconductance multipliers (Sec. 6-6) surpass them even in this respect.

(c) Magnetoresistance and Hall Effect Multipliers. The resistance of metallic bismuth can be varied over a 10:1 ratio through voltage-controlled magnetic-field changes between 0 and 10 kilogauss. This *magnetoresistance effect* has been utilized in feedback multipliers similar to Fig. 6-13*d*;[75,76] static accuracies within 0.1 percent of half-scale are possible.

In a suitable conductor carrying an excitation current i, a magnetic field H perpendicular to i produces a voltage

$$E_H = \lambda_H iB$$

across the conductor at right angles to both i and H, where B is the magnetic flux density due to H (*Hall effect*). Figure 6-13*e* shows a feedback multiplier utilizing the Hall effect. The gain-setting feedback loop controls the magnetic field for two similar thin semiconductor wafers (Hall generators) placed in the narrow air gap of a strong electromagnet. In practice, the semiconductor wafers may be sandwiched between iron or ferrite plates to reduce the air gap.

Even for the most sensitive Hall generators, the Hall coefficient λ_H is small (50 to 100 mV/kilogauss-ampere for thin indium-arsenide wafers), and the source impedance is low (of the order of 1 ohm) and therefore hard to match efficiently to a d-c amplifier. Lofgren[67] achieved a favorable compromise of Hall coefficient and source impedance with specially doped silicon wafers and obtained a static accuracy within 0.1 percent of half-scale (±20 volts), using a circuit similar to that of Fig. 6-13*e*. His paper describes circuit details, including a simple temperature-compensation method. It is also possible to compensate for the effects of asymmetrical contact placement.[69]

While the magnetoresistance and Hall effects work, in principle, at frequencies as high as several hundred KHz, the strong electromagnet required imposes severe bandwidth restrictions. Lofgren was able to trade flux density, and thus accuracy, for bandwidth by substituting a ferrite magnet core for the original ferromagnetic core, so that he obtained either 0.1 percent of half-scale static accuracy with full-scale output up to 20 Hz, or 0.3 percent of half-scale static accuracy with full-scale output up to 500 Hz (phase shift was not specified).[67] Hall-effect multiplication is used in special transducers, e.g., wattmeters.

(d) Modulation-type Multipliers. A variety of multiplication schemes depend on modulation of a sinusoidal or square-wave carrier waveform whose amplitude or mean absolute value is already proportional to one of the multiplier inputs; such modulation can be made more accurate with the feedback gain-control scheme of Fig. 6-12. Essentially every possible combination of amplitude modulation, pulse-width modulation, frequency modulation, phase modulation, and delta modulation has been tried.[77–110] In particular, relatively complicated *combined amplitude and frequency modulation (AM/FM) multipliers* have employed carrier frequencies as high as 2 MHz to obtain accuracy within 0.1 percent of half-scale at zero frequency and within 1 percent of half-scale up to 200 Hz,[79–81] but they could not maintain such accuracy for prolonged periods and were abandoned. The most successful dual-modulation scheme has been *amplitude/pulse-width modulation (AM/PWM; time-division multipliers*, Secs. 6-9 to 6-12), where the output product is obtained by simple averaging without complicated demodulator circuits. *Sampling multipliers*[108–110] can multiply a voltage by a *function* of a second voltage (Sec. 6-23). The bandwidths of all basic modulation-type multipliers are compromised by output low-pass filters needed to remove the carrier ripple (Sec. 6-12).

TIME-DIVISION MULTIPLIERS

6-9. Principle of Operation. *Time-division multipliers (AM/PWM multipliers, mark-space multipliers, pulsed-attenuator multipliers)* depend on combined pulse-amplitude and pulse-width modulation of a rectangular pulse train. The average (smoothed) values of the waveforms pictured in Fig. 6-14*a* and *b* are respectively given by

$$\frac{T_1}{T_1 + T_2} X = nT_1 X \qquad \frac{T_1 - T_2}{T_1 + T_2} X = (2nT_1 - 1)X \qquad \text{(6-10)}$$

where X is the pulse amplitude, and T_1 and T_2 are the widths of the positive and negative portions of the pulses; $n = 1/(T_1 + T_2)$ is the pulse

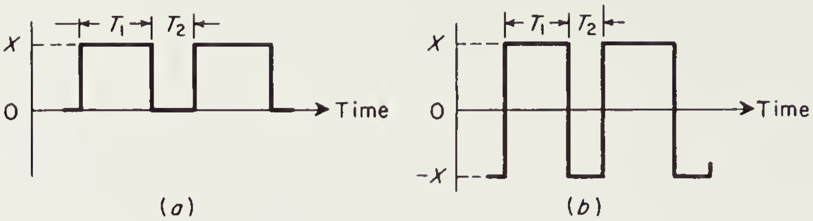


Fig. 6-14. The running-average value (d-c or low-frequency component) of the waveforms shown is $\frac{T_1}{T_1 + T_2} X$ in Fig. 6-14*a* and $\frac{T_1 - T_2}{T_1 + T_2} X$ in Fig. 6-14*b*.

repetition rate in hertz. These relations implement multiplication of a voltage X by a second voltage Y which controls the duty cycle nT_1 of the pulse train; the desired product is recovered by an averaging filter. In practice, the pulse repetition rate must be large compared with the highest signal frequency (Sec. 6-12). Repetition rates between 1 and 300 KHz are used.

Such multipliers yield high static accuracy (0.1 to 0.01 percent of half-scale) and fair frequency response (dynamic errors within 2 percent of half-scale for sinusoidal inputs up to $\frac{1}{25}$ of the pulse repetition rate used) at comparatively low cost. A complete four-quadrant multiplier of this type requires 6 to 14 transistors, in addition to three or four d-c amplifiers.

6-10. Externally Excited Time-division Multipliers. In Fig. 6-15, a phantastron or delay multivibrator is periodically triggered and pro-

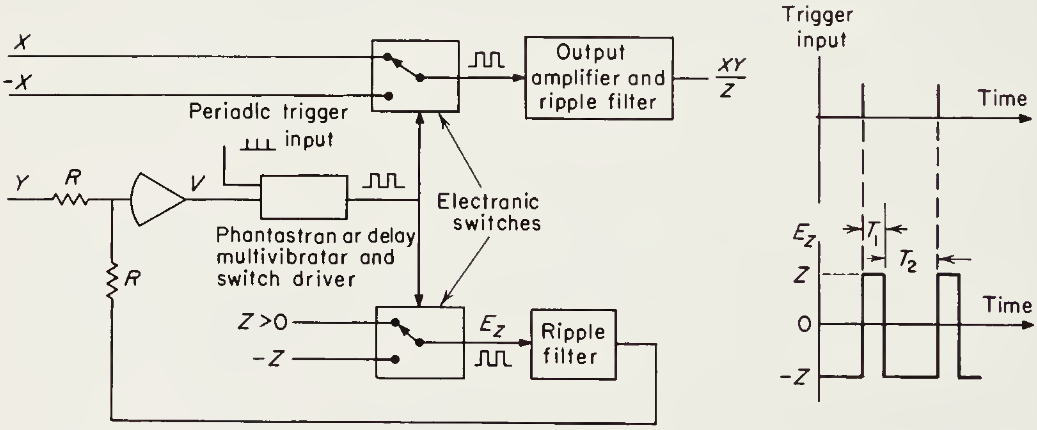


Fig. 6-15. Externally excited AM/PWM (time-division) multiplier. A periodically triggered phantastron or delay multivibrator generates gate pulses whose width increases with the voltage V produced by the gain-setting feedback loop. The circuit is capable of four-quadrant multiplication; the waveforms shown correspond to a positive input Y.

If one desires to employ electronic switches capable of switching only between Z or X and zero rather than between positive and negative inputs, one must use a biasing scheme similar to that of Fig. 6-16*b*.

duces rectangular gate pulses whose width is determined by the gain-setting input voltage V.* These gate pulses are suitably amplified and periodic triangular waveform is biased by the gain-controlling voltage V, so that a polarity-sensing comparator amplifier produces longer or shorter gate pulses, depending on the value of V. This multiplier circuit is capable of four-quadrant multiplication; the electronic switches used must work with inputs of either polarity. Additional switches may be driven synchronously to produce additional products by Y. The oscillators producing the triggering waveforms can be common to a number of multipliers.

6-11. Self-excited Time-division Multipliers. (a) Basic Four-quadrant Multiplier. The self-excited time-division-multiplier circuits of Fig. 6-16 oscillate of their own accord and do not require external trigger inputs.[97–99] In Fig. 6-16*a*, an initially positive switch output $E_Z = Z > 0$ combines with the multiplier input voltage Y to produce a positive integrator input current $Y/R_Y + Z/R_Z$ which is integrated downward. When the integrator output reaches the voltage E_{OFF}, the bistable multivibrator changes state and releases the electronic switches. E_Z is now equal to $-Z$, and the total integrator input current is $Y/R_Y - Z/R_Z$, which must be negative for proper operation, i.e.,

$$\frac{Z}{R_Z} - \frac{Y}{R_Y} > 0 \tag{6-11}$$

for all values of Y. The integrator output voltage then increases until it reaches the value E_{ON} required to return the bistable multivibrator to its original state. The process now repeats in a self-oscillatory fashion. It is seen that an increase in the multiplier input voltage Y accelerates the downward integration and slows the upward integration, so that the duty cycle nT_1 decreases.

More specifically, the integrator output voltage in Fig. 6-16*a* decreases at the rate $-\frac{1}{C}\left(\frac{Z}{R_Z} + \frac{Y}{R_Y}\right)$ and increases at the rate $\frac{1}{C}\left(\frac{Z}{R_Z} - \frac{Y}{R_Y}\right)$, so that

$$T_1 = \frac{(E_{\text{ON}} - E_{\text{OFF}})C}{Z/R_Z - Y/R_Y} \qquad T_2 = \frac{(E_{\text{ON}} - E_{\text{OFF}})C}{Z/R_Z + Y/R_Y} \tag{6-12}$$

Assuming perfect switching, the average value of the X-switch output equals the desired product

$$\frac{T_1 - T_2}{T_1 + T_2} X = -\frac{R_Z}{R_Y}\frac{XY}{Z} \tag{6-13}$$

---

* In such circuits, each trigger pulse starts a rectangular pulse which is integrated until the absolute value of the integral equals a predetermined fraction of the control voltage. At this point, the pulse is terminated by regenerative action.

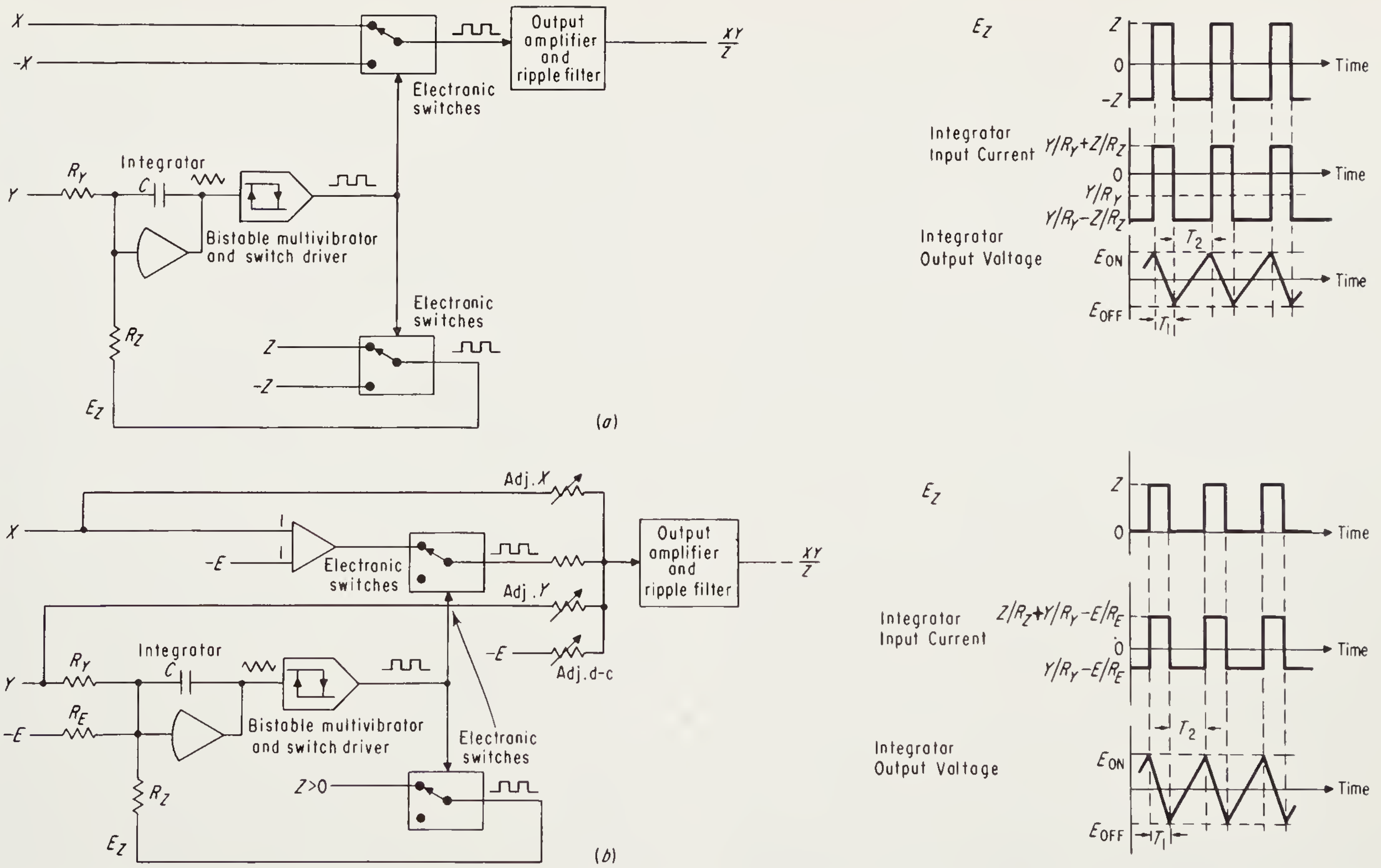


Fig. 6-16. Simplified block diagram of a self-excited time-division multiplier (*a*) and modified circuit (*b*) for use with simple single-throw electronic switches capable of passing only positive voltages. Four-quadrant multiplication is obtained through addition of a bias voltage $-E$ to both X and Y and subtraction of the unwanted product terms EX/Z, EY/Z, and E^2/Z in the output amplifier (see also Fig. 6-9*a*).

which is suitably scaled and inverted by the output amplifier to produce XY/Z. The pulse repetition rate is

$$n = \frac{1}{T_1 + T_2} = \frac{Z}{2(E_{\mathrm{ON}} - E_{\mathrm{OFF}})R_Z C}\left[1 - \left(\frac{R_Z}{R_Y}\frac{Y}{Z}\right)^2\right] \tag{6-14}$$

in hertz. Note that n decreases in the ratio $1 - (R_Z/R_Y)^2$ to 1 as Y changes from $Y = 0$ to $Y = \pm Z$.

Figure 6-16b illustrates a similar self-excited multiplier circuit designed to employ simpler (single-throw) electronic switches. This circuit will, in fact, yield four-quadrant operation even with electronic switches such as simple diode switches, which can pass current in one direction only (Sec. 5-2).[99]

(b) A Modified Circuit. The Telefunken* self-excited time-division multiplier employs an interesting modification of the circuit of Fig. 6-16 to make the pulse repetition rate less dependent on the reference input Z, so that direct division by Z becomes practical.[105,106] The modified circuit is shown in Fig. 6-17. Operation is similar to that in Fig. 6-16a, but the bistable multivibrator is replaced by a simple comparator (nominally without hysteresis), which receives positive feedback proportional to E_Z from the Z switch. The resulting regenerative loop again acts like a bistable multivibrator and changes states whenever the integrator output voltage decreases to $E_{\mathrm{OFF}} = -(R_1/R_2)Z$ or increases to $E_{\mathrm{ON}} = (R_1/R_2)Z$ (see also Sec. 4-12), assuming that any remaining hysteresis in the comparator circuit is negligible. The expression (13),

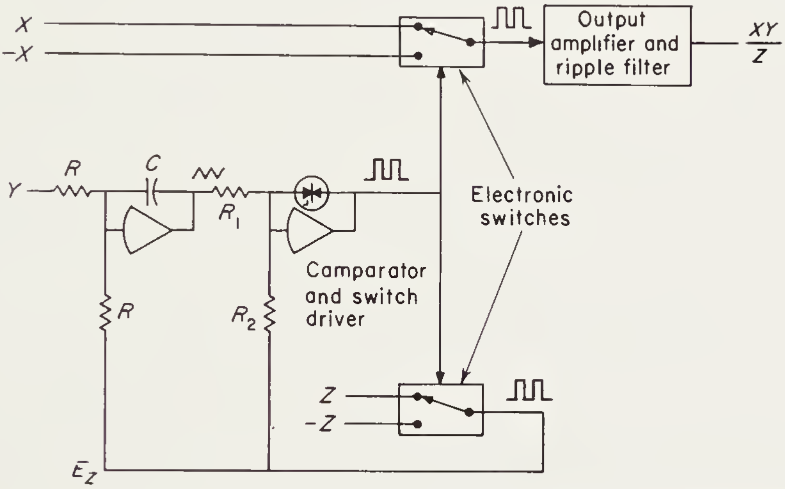

X
-X
Output amplifier and ripple filter
XY/Z
Y
R
C
R₁
Electronic switches
Comparator and switch driver
R
R₂
Z
-Z
E_Z

Fig. 6-17. In this improved self-excited time-division multiplier, the operate and release levels of the switch-actuating bistable circuit are proportional to the reference input Z. This reduces the dependence of the pulse repetition rate on Z, so that direct division by a variable input Z becomes practical. (*Telefunken G.m.b.H.*)

---

* Telefunken G.m.b.H., Konstanz, West Germany.

which determines the multiplier output for ideal switching, remains unchanged. The pulse repetition rate becomes

$$n = \frac{Z}{2(E_{\mathrm{ON}} - E_{\mathrm{OFF}})R_Z C}\left[1 - \left(\frac{R_Z}{R_Y}\frac{Y}{Z}\right)^2\right]$$

$$= \frac{1}{4R_Z C}\frac{R_2}{R_1}\left[1 - \left(\frac{R_Z}{R_Y}\frac{Y}{Z}\right)^2\right] \tag{6-15}$$

which is seen to depend much less on the reference input Z than the repetition rate (14) for the circuit of Fig. 6-16*a*.

As an example, let $R_Y = 2R_Z = 200\mathrm{K}$, $C = 1{,}000$ pF, and $Y = 10$ volts. Then in Fig. 6-16*a* with, say, $E_{\mathrm{ON}} - E_{\mathrm{OFF}} = 25$ volts, a decrease from $Z = 100$ volts to $Z = 10$ volts reduces n from about 20 to 1.5 KHz. But in Fig. 6-17 with $R_2/R_1 = 8$, a similar decrease in Z reduces n from about 20 KHz to only 15 KHz. As a result, the Telefunken circuit, which provides a kind of automatic gain control for the gain-setting feedback loop, permits direct division by a variable input Z over a wide range of values, a significant advantage (see also Sec. 6-16).

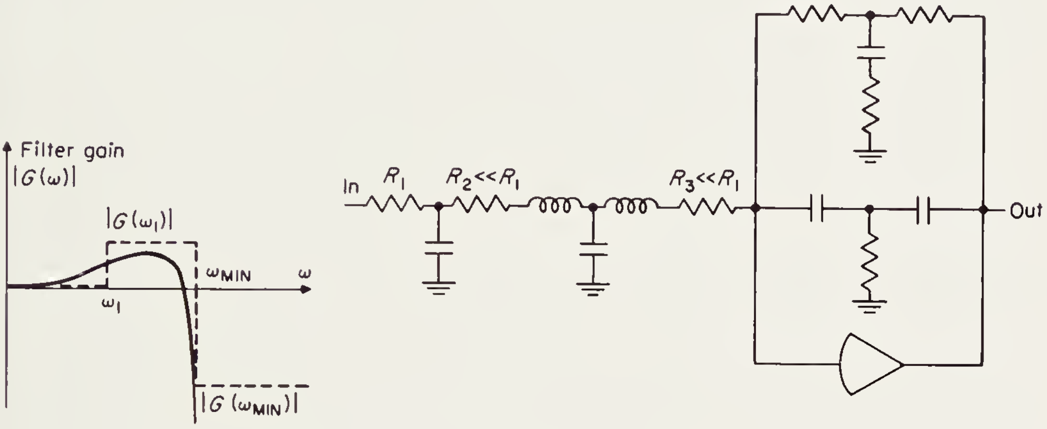
6-12. Static vs. Dynamic Accuracy in Multipliers Employing Ripple Filters. Ripple-filter Design. The design of every multiplier employing a high-frequency (1 KHz to 2 MHz) carrier involves an essential conflict between static multiplication accuracy, bandwidth, and carrier-ripple suppression. This problem, discussed here for time-division multipliers, is common to all modulation-type multipliers and applies also to triangle-averaging multipliers (Sec. 6-4).

Accurate multiplication requires accurate resistance networks, possibly mounted in common oil baths to maintain resistance ratios with temperature changes; low-drift chopper-stabilized operational amplifiers with high loop gains; and careful circuit layout and shielding to avoid crosstalk effects (noise and carrier beats) where several multipliers are mounted in close proximity to each other. Assuming also sufficient gain in the gain-setting loop, the remaining *static errors* in time-division multipliers will be due to:

1. *Imperfect switch timing*, i.e., asynchronous operation of the Z and X switches
2. *D-c offset variations* due to changing control-voltage leakage or rectified switching spikes in electronic switches (see also Secs. 5-1 and 5-5)
3. *Variations of switch timing, forward resistance, or back resistance* with the multiplier input voltage X or with tube aging

All these errors decrease with the ratio of the switching time to the repetition period,[100] so that lower carrier frequencies tend to yield improved static accuracy. But the frequency response of a time-division multiplier, like that of most modulation-type multipliers, is essentially that of the ripple filter needed to remove carrier-frequency components from the multiplier output, so that a compromise between static and dynamic accuracy must be made.

Phase shift and attenuation due to the ripple filter deteriorate multiplication accuracy at frequencies much lower than one-half the carrier frequency or repetition rate which, in principle, constitutes an upper frequency limit because of the sampled-data nature of the gain-setting loop. While moderate carrier-frequency noise is not usually objectionable at the input of integrators, summers, or recording devices, ripple voltages can be rectified by other multipliers or by function generators to cause low-frequency errors. Conventional low-pass ripple filters designed to reduce rms ripple below an acceptable 0.1 percent of full-scale would cause 3 to 5 deg phase shift at least two decades below the carrier frequency.



(a) (b)

Fig. 6-18. Frequency relationships for ripple-filter design (*a*) and filter design for a self-excited time-division multiplier (*b*). (*Based on Ref.* 105.)

Figure 6-18*a* illustrates the frequency relations determining ripple-filter design. The lowest carrier circular frequency $2\pi n$ cannot be higher than some value ω_{MIN} determined by the worst-case static errors obtained with a given switching circuit. If the amplitude response $|G(\omega)|$ of the ripple filter could have the ideal shape indicated in the dashed line of Fig. 6-18*a*, Bode's phase/amplitude relation for minimal-phase networks[105,106] indicates that the phase shift $\varphi(\omega)$ at the computing frequency $\omega/2\pi$ cannot be less than

$$\begin{aligned}\varphi(\omega) &= \frac{2\omega}{\pi}\int_0^\infty \frac{\log_e |G(\omega)|}{\omega^2}\,d\omega \\ &= \frac{2\omega}{\pi}\left[\frac{1}{\omega_{\text{MIN}}}\log_e |G(\omega_{\text{MIN}})| - \left(\frac{1}{\omega_1}-\frac{1}{\omega_{\text{MIN}}}\right)\log_e |G(\omega_1)|\right]\end{aligned} \tag{6-16}$$

where $|G(\omega_{\text{MIN}})|$ is the required ripple attenuation, and $|G(\omega_1)|$ is introduced to reduce the low-frequency phase shift (16) at the expense of an amplitude error at computing frequencies above $\omega_1/2\pi$. This amplitude error manifests itself, in particular, as overshoot and/or ringing in the multiplier step-input response. Real filters can only approximate the ideal filter characteristic (dashed line in Fig. 6-18*a*) in the manner shown by the solid line, so that amplitude errors start at lower computing frequencies. Practical filter designs will compromise between low phase shift and low overshoot, using the idealized relation (16) as a rough planning guide.

Figure 6-18*b* shows a practical filter design.[105] The operational-amplifier circuits used simplify the design by reducing interaction between the different filter sections. Fixed-carrier-frequency multipliers, such as externally excited time-division multipliers, can use band-suppression sections sharply tuned to the carrier frequency and its harmonics. This permits somewhat better ripple filtering than is possible for self-excited time-division multipliers, whose repetition rate varies by 15 to 30 percent as Y changes. Figure 6-18*b* shows a phase-compensating lead network in the output-amplifier feedback circuit, but it may be preferable to apply lead terms as a sort of high-frequency preemphasis at the multiplier inputs, so that the lead terms cannot enhance the output ripple.[23] Typical high-quality filter designs yield rms output ripple less than 0.1 percent of half-scale, and about 0.2 deg phase shift at $\omega_{\text{MIN}}/100$; the price paid for such low phase shift is, unfortunately, a step-function overshoot as large as 20 to 30 percent.

SOME IMPROVED MULTIPLICATION SCHEMES

6-13. High-frequency Correction of a Time-division Multiplier. Figure 6-19 shows a one-quadrant version of a feedback scheme suggested by Stojanovic and Milicevic to improve the bandwidth of an accurate time-division multiplier.[118] The output Z combines the output $XY + E$ of a relatively inexpensive wideband multiplier (incorporating an error E) and a feedback voltage $[K(XY/Z) - 1]$ derived from the accurate time-division multiplier, i.e.,

$$Z = XY + E + K\left(\frac{XY}{Z} - 1\right) \tag{6-17}$$

The feedback-amplifier gain K is large at low frequencies (where the time-division multiplier is at its best), so that then

$$Z \approx XY$$

At high frequencies, the gain K is rolled off to zero by the ripple filter, so that

$$Z = XY + E$$

i.e., the output is now mainly determined by the wideband multiplier and is not greatly affected by the ripple-filter phase shift. Operation of the nonlinear feedback circuit in the mid-frequency crossover range is not well understood and should be investigated with different ripple filters.

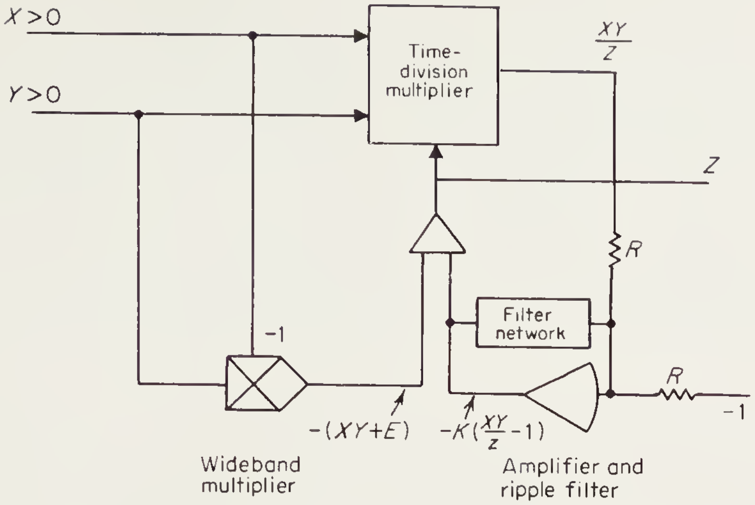


Fig. 6-19. A suggested method for correcting the high-frequency response of a time-division multiplier with an inexpensive wideband multiplier (scaling in machine variables). (*Based on Ref.* 118.)

6-14. Hybrid Analog-digital Multipliers Employing Coarse and Fine Channels.[115–117] If the residual errors of an electronic multiplier are not primarily due to aging, drift, or temperature effects but stay accurately constant, one should be able to improve the accuracy by adding a *small* correction function $f(X,Y)$ generated, say, by an inexpensive two-variable function generator, to the multiplier output (see also Sec. 4-19). Although the static errors encountered with most electronic multipliers are not sufficiently constant to justify this procedure, such correction schemes offer the possibility of improving *dynamic* errors at the expense of some circuit complexity; in particular, the correction-function generator may be a relatively inaccurate wideband electronic multiplier. Proper scaling will keep the correction function sufficiently small to prevent ill effects on static errors. A sophisticated and ingenious multiplier design due to Maslov[116] (Fig. 6-20) employs a wideband varistor quarter-square multiplier to correct the output of an accurate analog/digital/analog-converter multiplier. As a feature of special interest, the operation of this multiplier is predicated on the fact that, while the analog/digital/analog-converter multiplier (Sec. 6-8*a*) is relatively slow with respect to its Y input, it has essentially the bandwidth of an operational amplifier with respect to the X input. As a result, the coarse varistor multiplier usually has to supply only a relatively small product $-\alpha X_1X_2$ due to high-frequency components of X and Y. The correction terms X_2, Y_2 in Fig. 6-20*b* are, however, automatically adjusted to compensate for errors in the A/D/A conversion. If, say, Y varies too rapidly for accurate conversion, the correction Y_2 takes up the slack, and correct multiplication continues at

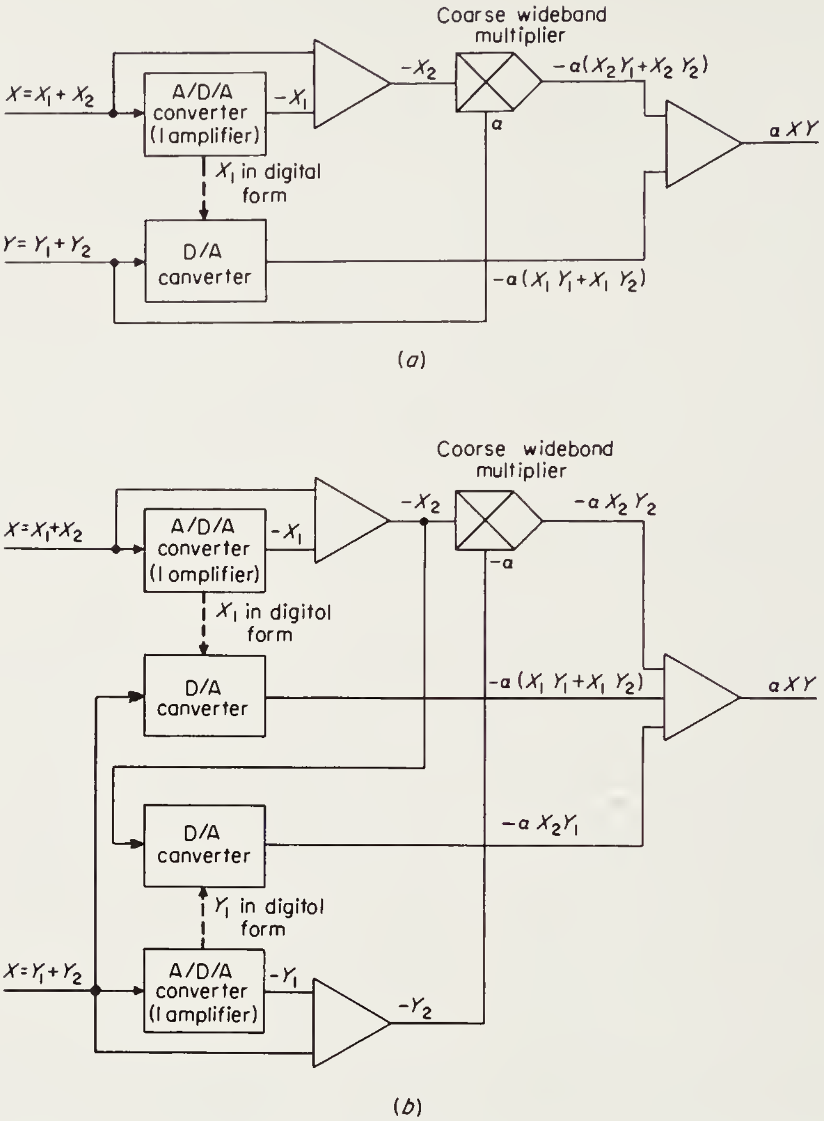


Fig. 6-20. Multiplier circuits designed to combine products from "coarse" and "fine" channels.

reduced accuracy. This is true even if the *Y*-channel A/D/A converter fails completely and produces an erroneous output voltage. A simple deadspace circuit (back-to-back junction diodes) following the coarse multiplier prevents the latter from contributing excessive errors when the product output is near zero.

Maslov's original multiplier used simple stepping-switch converters and was able to combine static accuracy within 0.02 percent of half-scale with 1 percent multiplication at 10 KHz, a remarkable result. An improved multichannel multiplier could employ all-electronic A/D/A conversion and a fast transconductance multiplier.

ELECTRONIC MULTIPLIERS: TESTING AND DIVISION CIRCUITS

6-15. Multiplier Adjustment and Testing. The static multiplication error of an analog multiplier can be arbitrarily written in the form

$$e(X, Y) = X_0 - \alpha XY = e_0 + e_X X + e_Y Y + e_{XY}(X, Y)XY \quad (6\text{-}18)$$

where αXY is the desired multiplier output voltage, and e_0, e_X, e_Y are approximately constant for small X, Y. The following adjustment procedure will reduce the first three static-error terms on the right of Eq. (18):

1. With $X = Y = 0$, change e_0 by adding a d-c voltage to the multiplier output until $X_0 = e_0 = 0$. This adjustment minimizes the *static zero error.*
2. With $Y = 0$, $X = X_{\text{max}}$ (100 volts, say), change e_X by adding a voltage proportional to X to the multiplier output until $X_0 = e_X X_{\text{max}} = 0$.
3. With $X = 0$, $Y = Y_{\text{max}}$, change e_Y by adding a voltage proportional to Y to the multiplier output until $X_0 = e_Y Y_{\text{max}} = 0$.
4. With $X = X_{\text{max}}$, $Y = Y_{\text{max}}$, adjust the actual multiplier gain until $e(X_{\text{max}}, Y_{\text{max}}) - \alpha XY = e_r(X_{\text{max}}, Y_{\text{max}}) = 0$.

These adjustments may interact slightly and should be repeated until no further change is noted. Since e_X and e_Y may depend somewhat on X and Y, respectively, it may be possible to reduce static errors by a compromise between the above adjustments and similar adjustments with $X = -X_{\text{max}}$ and/or $Y = -Y_{\text{max}}$. Note the readjustment of the static zero error (step 1), since errors near $X = 0$, $Y = 0$ will cause large absolute percentage errors.

Figure 6-21 shows a practical switching circuit designed to simplify the adjustment procedure. Such a calibration circuit can be shared by several multiplier channels. In most of the electronic-multiplier types discussed, the d-c, X, and Y adjustments are simply trimming resistors in an output-amplifier summing network.

A possibly better, if slightly more complex, method for testing and calibrating analog multipliers is shown in Fig. 6-22. Here the input voltage Y is set to $Y = -1/\alpha$, while a triangular voltage waveform sweeps the other input X through its full range. The error voltage

$$X_0 + X = X_0 - \alpha XY$$

is slightly filtered to remove high-frequency noise and/or ripple, amplified, and continuously displayed on an oscilloscope or recording device. With a low-frequency triangular waveform (0.1 to 2 Hz), static-error adjustments are readily made by reference to the output

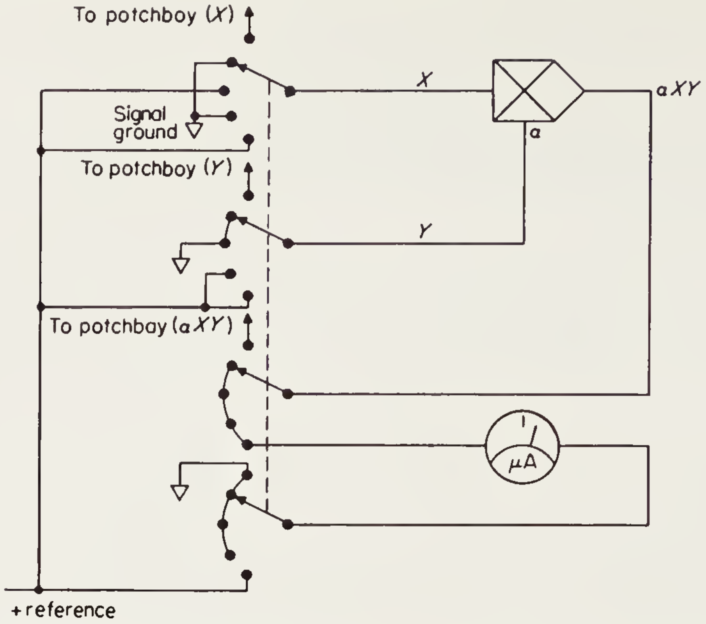


Fig. 6-21. A multiplier calibration circuit. Such a circuit can be shared by four to eight multiplier channels.

display. The roles of the two inputs can be interchanged. Dynamic errors are checked with the same computer setup with higher input frequencies. In multipurpose computers, the triangular waveform is easily generated by the computer itself (Sec. 4-12). The entire test setup is often included in prewired test boards for electronic analog computers.

Small-signal amplitude and phase response of a multiplier can be measured with, say, $X = 10$, $Y = 10 \cos \omega\tau$.

The maximum of the *static residual multiplication error* $e_r(X, Y)XY$ remaining after careful adjustment is the static error quoted in most manufacturers' specifications. Since practical static errors are largely due to *changes* in the semiconductors and voltage levels after calibration, honest specifications ought to state the maximum static error 8 hr and/or 1 week after adjustment as well as immediately after calibration.

6-16. Electronic Division. Electronic division is most frequently implemented through implicit solution of the equation $\alpha X_0 Y + X = 0$

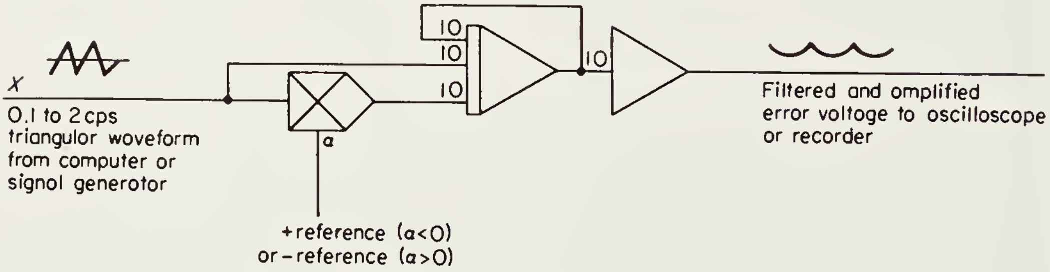

X
0.1 to 2 cps triangular waveform from computer or signal generator
a
+reference (a<0) or −reference (a>0)
10
10
10
10
Filtered and amplified error voltage to oscilloscope or recorder

Fig. 6-22. Triangle-input circuit for multiplier testing and adjustment.

with a multiplier and high-gain amplifier in the manner of Figs. 1-2 and 6-23, so that $X_0 = -X/Y$; Y must not change sign (see also Secs. 6-7 and 8-9*b*). This division technique applies to all types of multipliers; the high-gain feedback loop may require special stabilization networks for smooth loop-gain rolloff, particularly if the multiplier in question has a sharp-cutoff ripple filter (Fig. 6-23). Note also that the divider accuracy, frequency response, and output noise will all vary with the input Y, since the latter affects the loop gain. Automatic gain control in the division loop can improve this situation, but the added cost is usually not justified, and a worst-case design over a suitable range of Y values suffices.

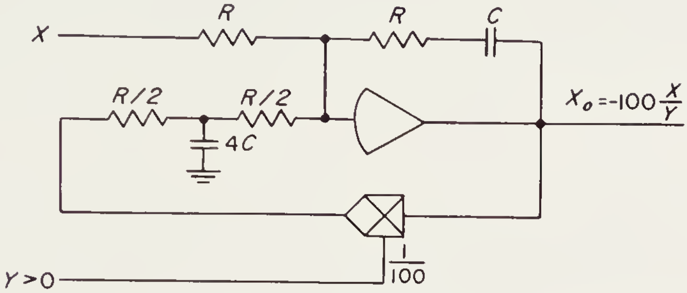


Fig. 6-23. A possible equalization scheme for high-gain division loops employing electronic multipliers with ripple filters.[23]

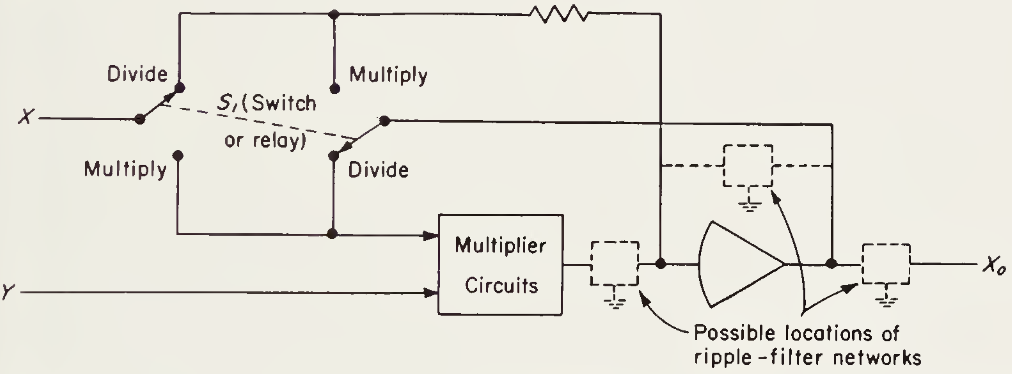


Fig. 6-24. Electronic multiplier with multiply-divide mode switch or relay. S_1 switches the input X and converts the multiplier output amplifier into a high-gain amplifier for division by Y. Additional switch or relay contacts can be used to introduce various stabilization networks into the division loop.

In many electronic multipliers, a switch or patchbay-controlled relay can reconnect the multiplier output amplifier as a high-gain amplifier for division (DIVIDE *mode*, Fig. 6-24), so that no external amplifier is required. Additional contacts on the multiply-divide switch can also change the amplifier equalization as needed.

Since in our feedback division scheme small output voltages X_0 correspond to larger products $X_0 Y$, multipliers whose output error decreases with the product output perform best in division loops. This is true for time-division multipliers but not for diode quarter-square circuits, whose output error fluctuates more or less periodically with increasing input voltages.

In principle, every multiplier employing the feedback gain-control scheme of Fig. 6-12 can serve for division by its reference input Z, which can be variable as well as fixed. This division scheme conveniently produces the output $\alpha XY/Z$ but usually is employed only in special-purpose computers where Z varies over a relatively small voltage range; most designers prefer a constant loop gain in the critical multiplier gain-control loop (Fig. 6-12). Automatic gain control can, again, improve this situation. In particular, the Telefunken multiplier circuit of Fig. 6-17 inherently has automatic gain control in its gain-setting loop and permits direct division by Z.[105,106]

VECTOR RESOLUTION AND COMPOSITION

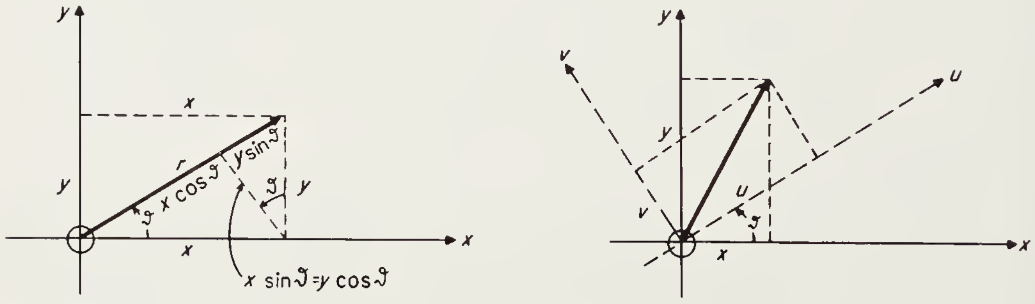
6-17. Vector Resolution and Composition. Important computer applications, such as navigation, range instrumentation, trajectory computation, and flight simulation require one to implement relations of the the form

$$x = r\cos\vartheta \qquad y = r\sin\vartheta \tag{6-19}$$

Equation (19) represents, in particular, the *resolution of a vector of magnitude r along rectangular cartesian axes*, or the *transformation* from polar point coordinates r, ϑ to rectangular *cartesian coordinates* x, y (Fig. 6-25*a*). Such relations are also combined in the transformation

$$u = x\cos\vartheta + y\sin\vartheta \qquad v = -x\sin\vartheta + y\cos\vartheta \tag{6-20}$$

which represents either (1) the new components (coordinates) u, v of a given vector in a u, v coordinate system rotated through the positive angle ϑ with respect to the x, y system or (2) the components u, v of a vector rotated through the angle $-\vartheta$ (Fig. 6-25*b*). Three-dimensional rotations are described by similar somewhat more complicated formulas.[126] The great practical importance of such relationships has led to the development of multiplier/function generators known as *resolvers*, which are specifically designed to produce products of the form (19).



(*a*) (*b*)

Fig. 6-25. Resolution of a vector in polar coordinates (*a*) and transformation of vector components or coordinates (*b*).

Vector composition (*inverse resolution*, rectangular-to-polar coordinate conversion) requires computer setups which produce outputs corresponding to

$$r = \sqrt{x^2 + y^2} \tag{6-21a}$$

$$\vartheta = \arctan \frac{y}{x} \qquad (x^2 + y^2 \neq 0) \tag{6-21b}$$

for pairs of inputs proportional to x and y. Although Eq. (21*a*) can be implemented directly, one more frequently obtains ϑ through implicit solution of the equation

$$-x \sin \vartheta + y \cos \vartheta = 0 \tag{6-22a}$$

so that r can be produced in the form

$$r = x \cos \vartheta + y \sin \vartheta \tag{6-22b}$$

(Fig. 6-25*a*; see also Sec. 6-22). Commercially available resolvers usually yield two pairs of products ($\pm X \sin \vartheta$, $\pm X \cos \vartheta$, $\pm Y \sin \vartheta$, $\pm Y \cos \vartheta$) and can be set up to implement Eq. (22) (*inverse-resolver* or POLAR mode of operation) as well as Eqs. (19) and (20) (*resolver* or RECTANGULAR mode).

6-18. Resolver Servos Using Sine-Cosine Potentiometers. Figure 6-26*a* shows a *resolver servomechanism* positioning precision continuous-rotation sine-cosine potentiometers in accordance with the input voltage $a_\vartheta \vartheta$. Such servo resolvers can have static accuracies within 0.05 percent of full scale (200 volts) for 2 volts/deg scaling. Film-type sine-cosine

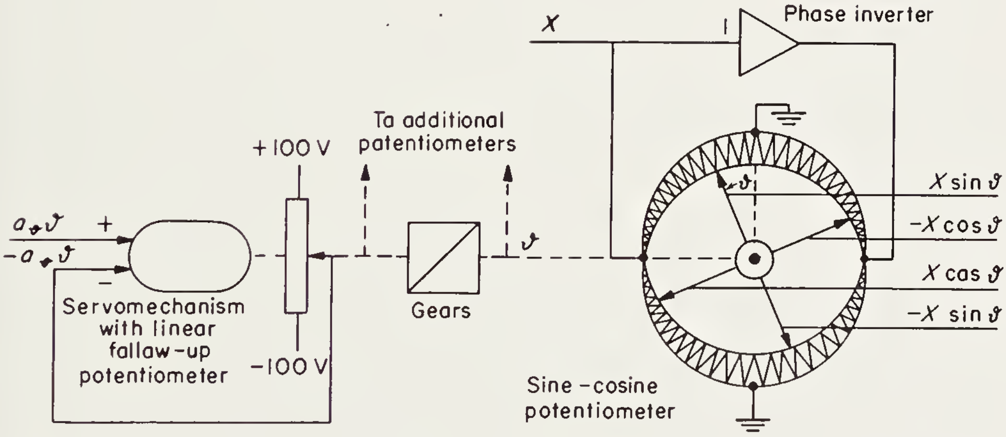


Fig. 6-26*a*. Basic resolver servo employing wirewound continuous-rotation sine-cosine potentiometers. The sine-cosine potentiometer is geared to the (1-turn or 10-turn) linear follow-up potentiometer, so that the resolver-potentiometer shaft turns through ϑ deg corresponding to $a_\vartheta \vartheta$ volts follow-up voltage. In ±100-volt computers, the scale factor a_ϑ is usually ½ volt/deg ($-200 \text{ deg} \leq \vartheta \leq 200$ deg) or ¼ volt/deg ($-400 \text{ deg} \leq \vartheta \leq 400$ deg). It is possible to use end resistances on the follow-up potentiometer to obtain, instead, a ±180-deg or ±360-deg range with slightly improved accuracy.

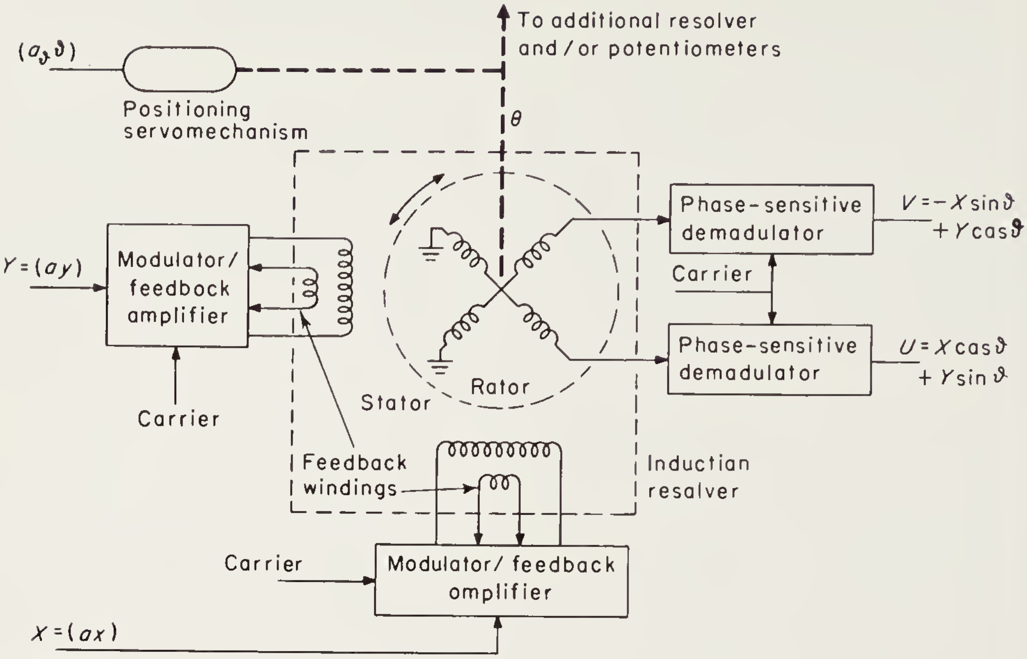

To additional resolver and/or potentiometers
(aϑ)
Positioning servomechanism
θ
Y=(ay)
Modulator/ feedback amplifier
Carrier
Stator
Rotor
Feedback windings
Phase-sensitive demodulator
V=−X sin ϑ + Y cos ϑ
Carrier
Phase-sensitive demodulator
U=X cos ϑ + Y sin ϑ
Induction resolver
Carrier
Modulator/ feedback amplifier
X=(ax)

Fig. 6-26*b*. Servo resolver employing an a-c induction resolver with precision modulators and demodulators to implement the coordinate-rotation transformation (23). For $X = 0$, one obtains the simpler outputs $Y \cos \vartheta$, $Y \sin \vartheta$. Many induction resolvers have push-pull rotor outputs instead of the single-ended outputs indicated here.

potentiometers permit greater servo bandwidth with 0.3 percent static accuracy. Different gear ratios permit 4 volts/deg scaling with $-360 \text{ deg} \leq \vartheta \leq 360 \text{ deg}$ at reduced accuracy; the range of ϑ can, instead, be extended with relay or diode quadrant-switching circuits (Sec. 6-21). Servo resolvers used in general-purpose analog computers usually have two sine-cosine potentiometers and two or more linear potentiometers for follow-up and multiplication. Their sine-cosine potentiometers are usually designed for 1M load resistances.

6-19. Induction Resolvers. *Induction resolvers* may be regarded as special a-c transformers with two mutually perpendicular *stator windings* and two mutually perpendicular *rotor windings* rotated by a servo shaft. If modulated a-c voltages $X \sin \omega\tau$, $Y \sin \omega\tau$ are applied to the resolver stator windings, the two rotor output voltages are $U \sin \omega\tau$, $V \sin \omega\tau$, with

$$U = X \cos \vartheta + Y \sin \vartheta \qquad V = -X \sin \vartheta + Y \cos \vartheta \tag{6-23}$$

where ϑ is the servo-shaft displacement, so that Eq. (19) or Eq. (20) can be implemented with the aid of phase-sensitive modulators and demodulators (Fig. 6-26*b*). Note that carrier phase reversal corresponds to a sign change in the modulating waveform. The most frequently employed carrier frequency $\omega/2\pi$ is 400 Hz, which permits the modulation waveforms $X(\tau)$, $Y(\tau)$, $U(\tau)$, $V(\tau)$ a useful bandwidth between 0 and 5 to 15 Hz. As a rule, a square-wave carrier rather than a sinusoidal carrier is used in order to simplify modulation and demodulation.

The design of accurate induction resolvers is described in Refs. 4 and 5. High-quality induction resolvers have at least as good conformity as sine-cosine potentiometers. Induction resolvers have infinite resolution and, since their only rotating contacts are sliprings, there is no wear on precision parts; for the same reason, one obtains lower output noise, reduced friction, and a longer maintenance-free life. The main disadvantage of a-c resolvers is the requirement for very accurate modulation and demodulation; in addition, the carrier phase must be carefully controlled throughout each resolver system to avoid errors in addition and/or demodulation.

In special-purpose computers, several operations on trigonometric functions can be combined without intermediate demodulation and modulation.

6-20. Electronic Resolvers Combining Function Generators and Multipliers. Accurate resolver servos are both expensive and slow. Accurate all-electronic implementation of the multiple products by trigonometric functions required especially in aircraft and space-vehicle simulation still represents a challenging and somewhat difficult problem because of the possible error accumulation in differences of products. Conventional analog-computer setups for three-dimensional coordinate transformations are all cumbersome and expensive; it is to be hoped that more elegant and more easily serviced hybrid analog-digital techniques will be developed in the future (see also Sec. 9-8).

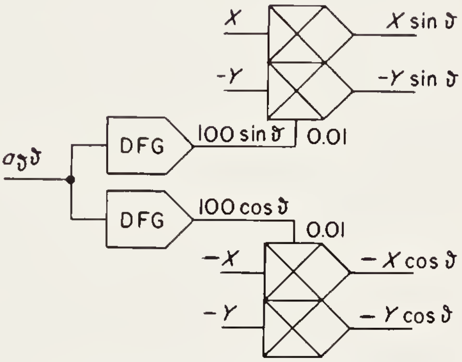


Fig. 6-27. Electronic resolver using diode function generators and electronic multipliers.

The most straightforward electronic resolver circuits simply combine the best available diode function generators with electronic multipliers (Figs. 6-27 and 6-28). A suitable patchbay layout can permit different connections, such as POLAR and RECTANGULAR modes, complete Euler-angle transformations, and individual use of the electronic multipliers and function generators.[125–139]

6-21. Quadrant Switching and Rate Resolvers. **(a)** Although many sine-cosine potentiometers (Fig. 6-26*a*) permit continuous rotation and hence do not limit the range of the resolver angle ϑ, the resolver-servo follow-up potentiometer limits the range of ϑ within fixed limits, usually ±90 deg, ±180 deg, or ±360 deg. Diode function generators, like that in Fig. 6-28, are usually restricted to $|\vartheta| \leq 90$ deg. In applications requiring a wider range of angles (e.g., tumbling aircraft motion, special maneuvering turns), one can implement relations like

$$\left.\begin{aligned} \sin\left(\pm\frac{\pi}{2}+\varphi\right) &= \sin\left(\pm\frac{\pi}{2}-\varphi\right) \\ \sin\left(\pm\frac{3\pi}{2}+\varphi\right) &= \sin\left(\pm\frac{3\pi}{2}-\varphi\right)\ldots \end{aligned}\right\} \qquad (6\text{-}24a)$$

$$\cos(-\varphi) = \cos\varphi \qquad \cos(\pm\pi+\varphi) = \cos(\pm\pi-\varphi) \qquad (6\text{-}24b)$$

with *quadrant-switching circuits* actuated whenever ϑ reaches predetermined limits. A quadrant switch intended to extend the range of a

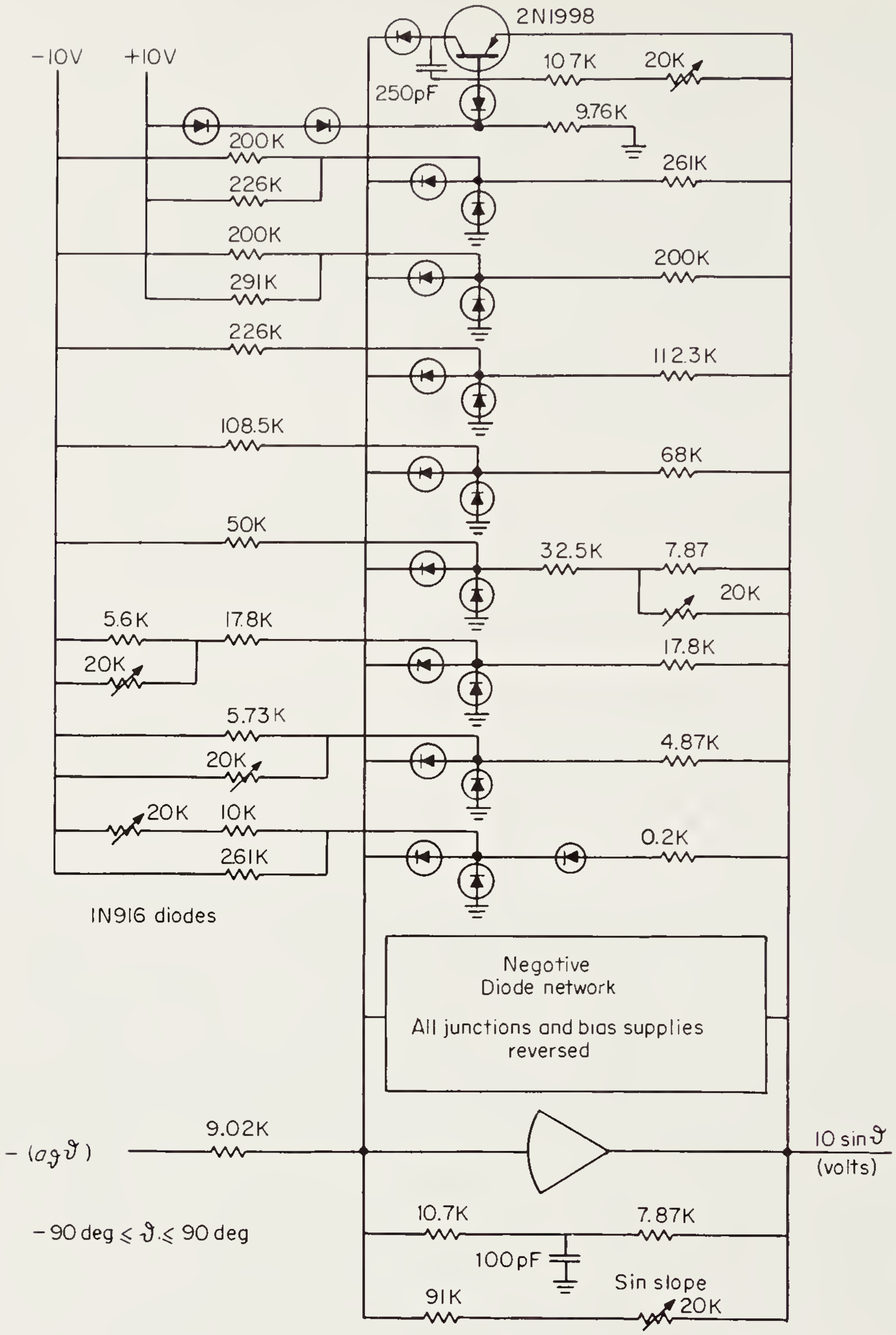


Fig. 6-28. A wideband ±10-volt sine-function generator. Successive feedback-limiter channels, and finally a pair of transistor limiters (Sec. 4-8), reduce the slope of 10 sin θ to zero as $|\theta|$ increases to 90 deg; combined series-shunt diode limiting (Sec. 4-5) improves high-frequency operation; errors are within 0.2 percent of half-scale at zero frequency and are still 0.5 percent of half-scale at 10 KHz. (*University of Arizona.*)

± 90-deg sine-function generator to $+270$ deg would, for instance, replace the function-generator input $a_\vartheta \vartheta$ by $a_\vartheta(\pi - \vartheta)$ whenever ϑ exceeds $\pi/2$. Figure 6-29 shows a number of electronic quadrant-switching schemes which are readily implemented with the accurate precision-limiter circuits of Sec. 4-10; static accuracies well within 0.1 percent of half-scale are possible.

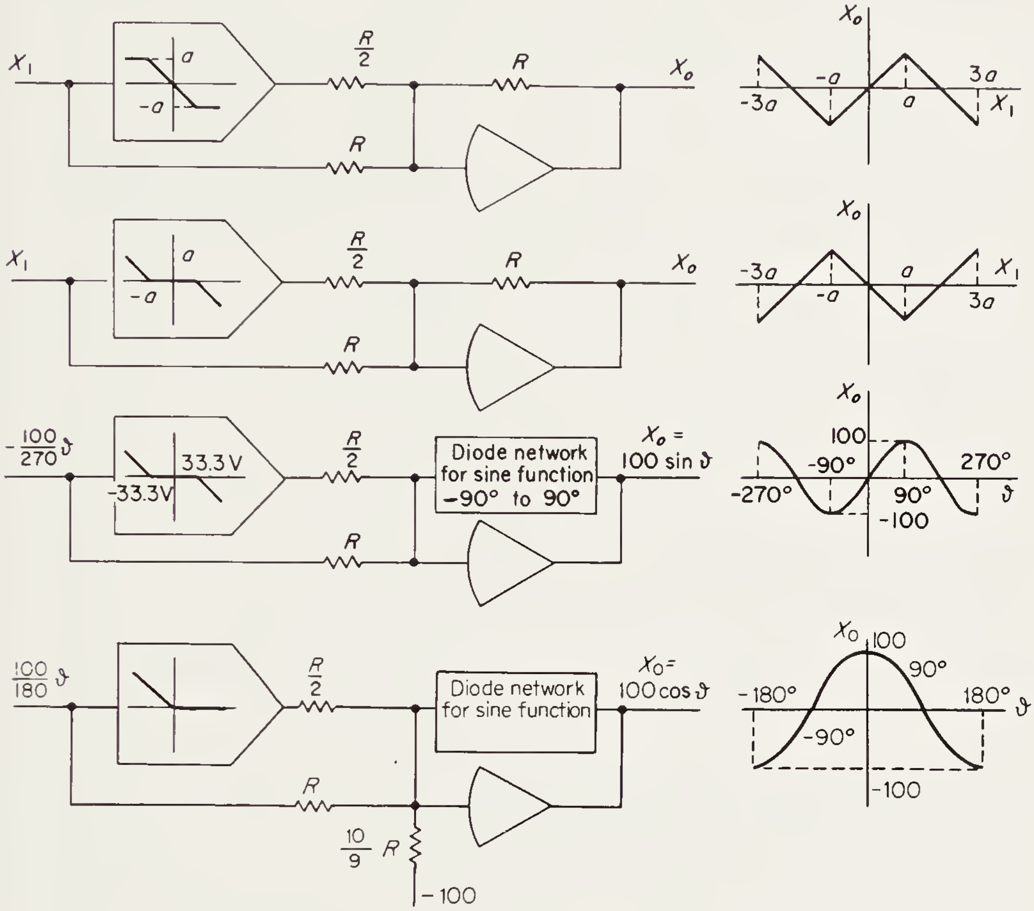


Fig. 6-29. Electronic quadrant-switching circuits used to extend the range of trigonometric-function generators. The precision-limiter circuits of Sec. 4-10 are particularly useful in this application.

(b) In many applications, a voltage proportional to the time derivative $d\vartheta/d\tau$ of the desired resolver angle ϑ is available in the computer setup. In this case, we can, if desired, obtain unlimited resolver rotation by driving a 360-deg electromechanical resolver with a *rate servo* whose output-shaft displacement ϑ changes at a rate proportional to the input voltage $d\vartheta/d\tau$, so that

$$\vartheta = \int_0^\tau \frac{d\vartheta}{d\tau}\,d\tau + \vartheta(0) \tag{6-25}$$

In such a *rate resolver*, the follow-up potentiometer feedback is replaced by d-c tachometer feedback. Since the resolver now acts as an integrator as well, we shall require control relays for resetting to initial conditions [$\vartheta = \vartheta(0)$, see also Sec. 8-9*b*]; the servo feedback must be switched from a rate signal to a position follow-up in RESET. The follow-up potentiometer used with a resolver having a rate-resolver mode must be a single-turn potentiometer bridged to permit continuous rotation; either solenoid-retractable limit stops or switched electrical limit stops may have to be provided. The design and equalization of computer servomechanisms with rate feedback are discussed in some detail in Refs. 1 to 5.

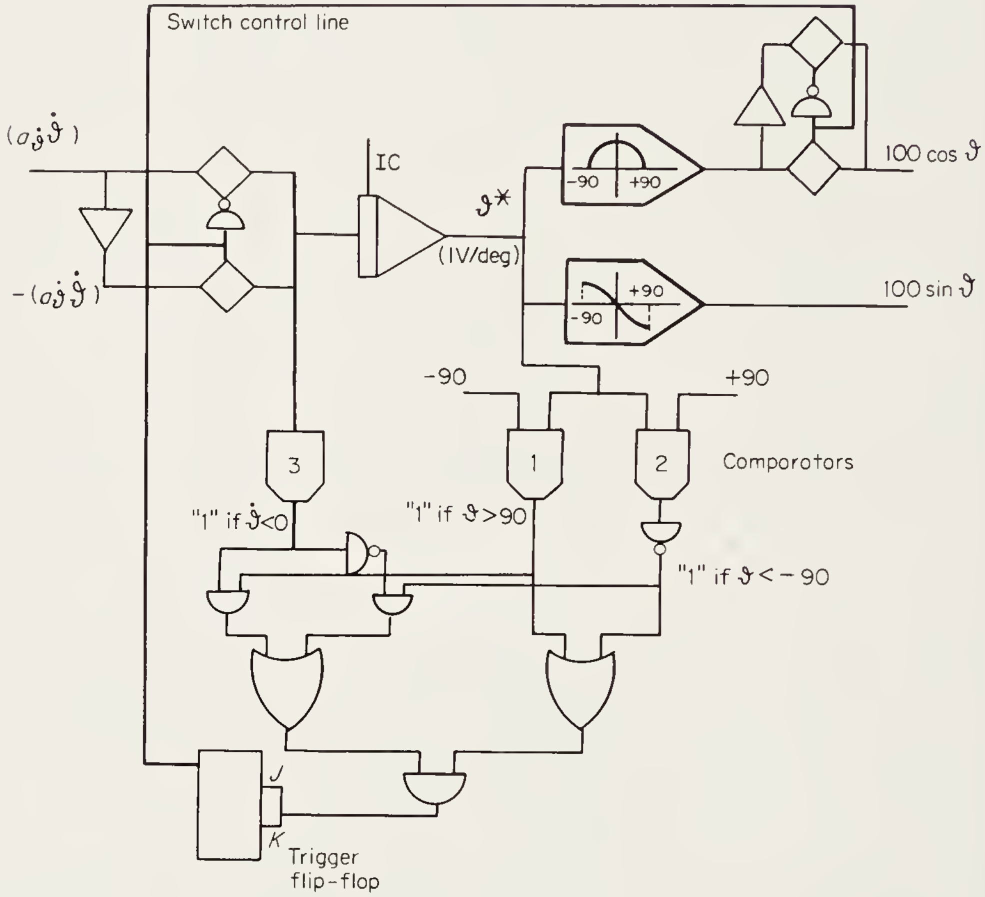


Fig. 6-30. An electronic rate resolver generating 100 sin θ and 100 cos θ with the aid of electronic single-pole/double-throw switches controlled by comparators and logic. Comparators 1 and 2 are the quadrant-sensing comparators. Comparator 3 and the associated logic prevent errors due to sign changes in θ. The trigger flip-flop triggers on logic transitions to 1. Scaling shown is in volts. (*Refs.* 134 *and* 136.)

Figure 6-30 illustrates a typical *automatic quadrant-switching scheme for an all-electronic rate resolver* (see also Sec. 6-24).

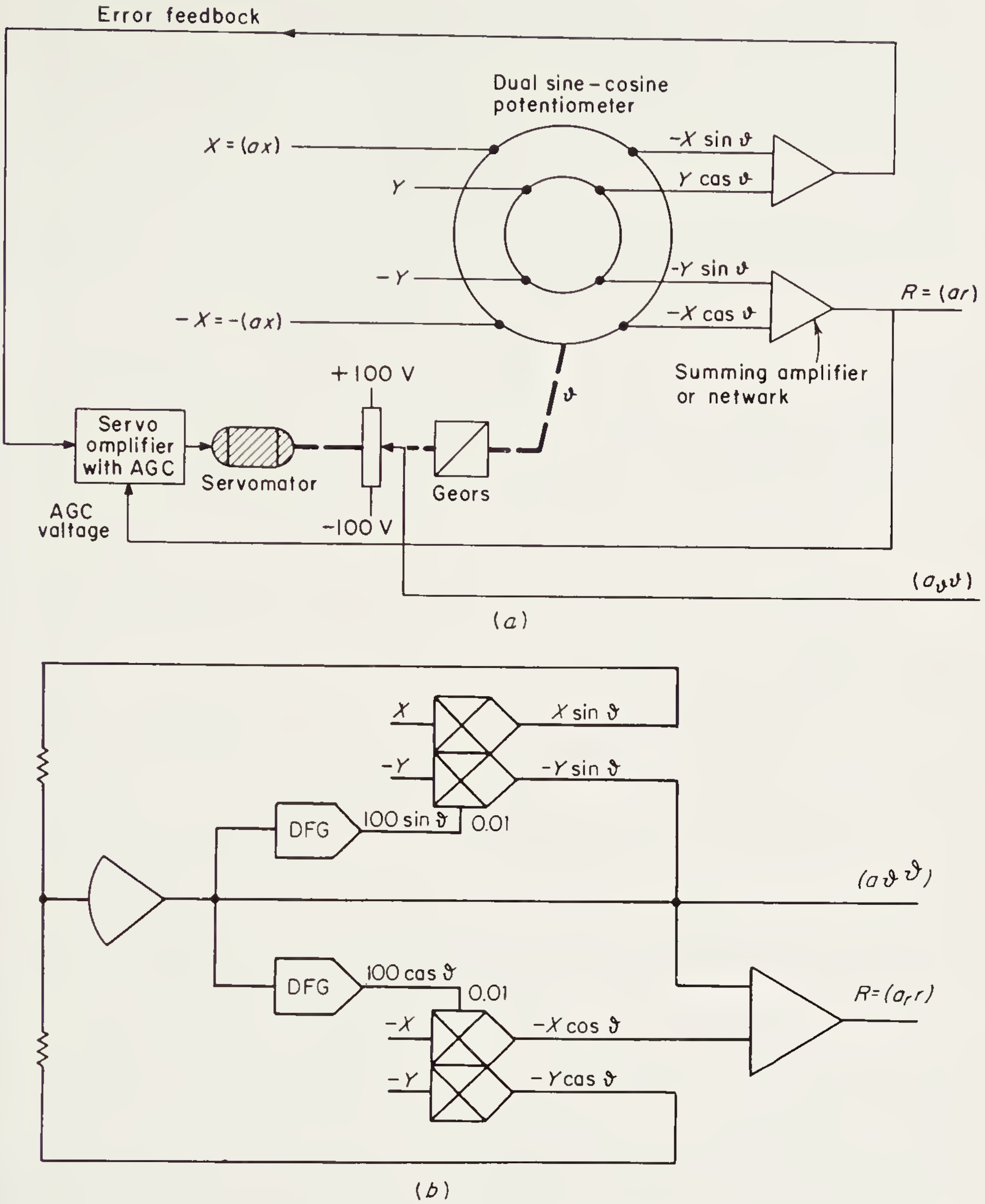


Fig. 6-31. Rectangular-to-polar coordinate transformation (POLAR mode of resolver, inverse resolver) using a servo resolver (*a*) and electronic resolver circuits (*b*).

6-22. Inverse Resolvers (POLAR Resolver Mode). Figure 6-31 shows a resolver-servo connection implementing Eq. (22) to yield voltages proportional to r and ϑ when the rectangular coordinates or vector components x, y are given. The feedback loop will, in each case, null an error signal proportional to $x \sin \vartheta - y \cos \vartheta$. The resulting feedback ratio is proportional to $r = \sqrt{x^2 + y^2}$, and $\vartheta = \arctan (y/x)$ is necessarily indeterminate for $x = y = 0$. To improve the servo performance, we can employ automatic gain control (AGC) to increase the servo-amplifier gain for small r, as in Fig. 6-31a. Practical AGC circuits (Refs. 2 and 134) will keep the loop gain within about 6 db as r varies over a 50:1 to 100:1 range.

If one requires only an output proportional to r, it may be simpler to implement Eq. (21a).

6-23. Use of Sampling and Comparator Circuits. **(a)** The *sample-hold resolver* shown in Fig. 6-32 is based on the principle of the sampling multiplier of Ref. 108. A carefully adjusted repetitive-analog-computer setup (shaded integrators in Fig. 6-32) generates the voltages $-R \sin \omega\tau$, $-R \cos \omega\tau$, and $(b/\pi)(\omega\tau - \pi)$ 20 to 1,000 times per second. A fast comparator samples $-R \sin \omega\tau$ and $-R \cos \omega\tau$ whenever the periodic ramp voltage $(b/\pi)(\omega\tau - \pi)$ reaches the value of the input voltage $(a\vartheta)$. The filtered sample-hold output voltages are, then, $R \sin (a\pi/b)\vartheta$ and $R \cos (a\pi/b)\vartheta$, where both $(a\vartheta)$ and the initial-condition input R can be slowly varying machine variables.[109,110]

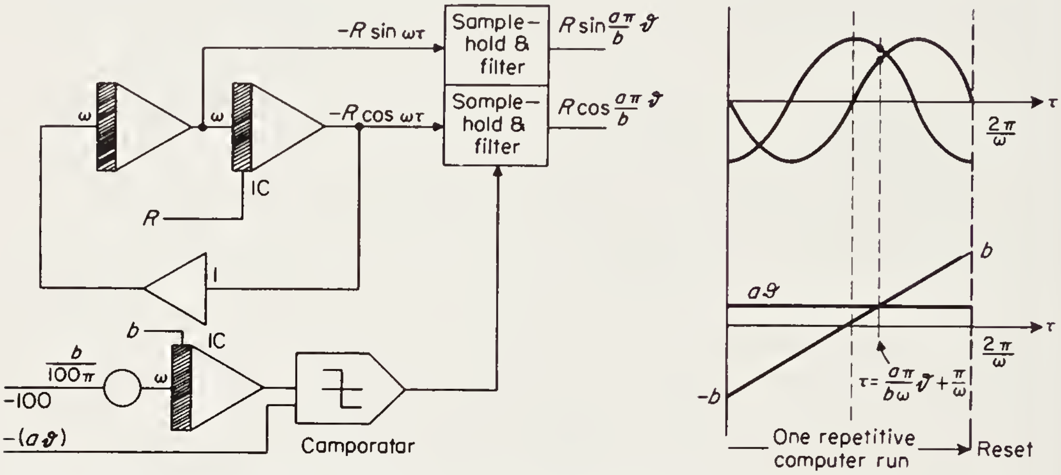


Fig. 6-32. A sample-hold resolver using repetitive-analog-computer circuitry. Shaded integrators are under rep-op control. First-order holds are recommended (Sec. 9-3).

A sample-hold resolver performs both sine-cosine generation and multiplication in one operation but is subject to a severe compromise between computing bandwidth (about 1/628 of the repetition rate for, say, 1 percent zero-order-hold error) and accuracy. For an accuracy within ϵ percent of half-scale, the worst comparator/switch timing error must be below $\epsilon/2\pi$ percent of one repetitive-computer-run period $2\pi/\omega$ so that, for example, a 10-μsec timing error limits the repetition rate to 140ϵ runs per second if we allow for a 10 percent reset period after each run. The corresponding computer bandwidth would be less than 6ϵ Hz.

(b) The circuit of Fig. 6-33a[129-131] employs a pulse-width modulator with sinusoidal excitation to generate periodic pulses of width $(\pi - 2X)/\omega$ symmetrical with respect to the sine-wave peaks, where X is measured in radians ($|X| < \pi/2$). These pulses gate the

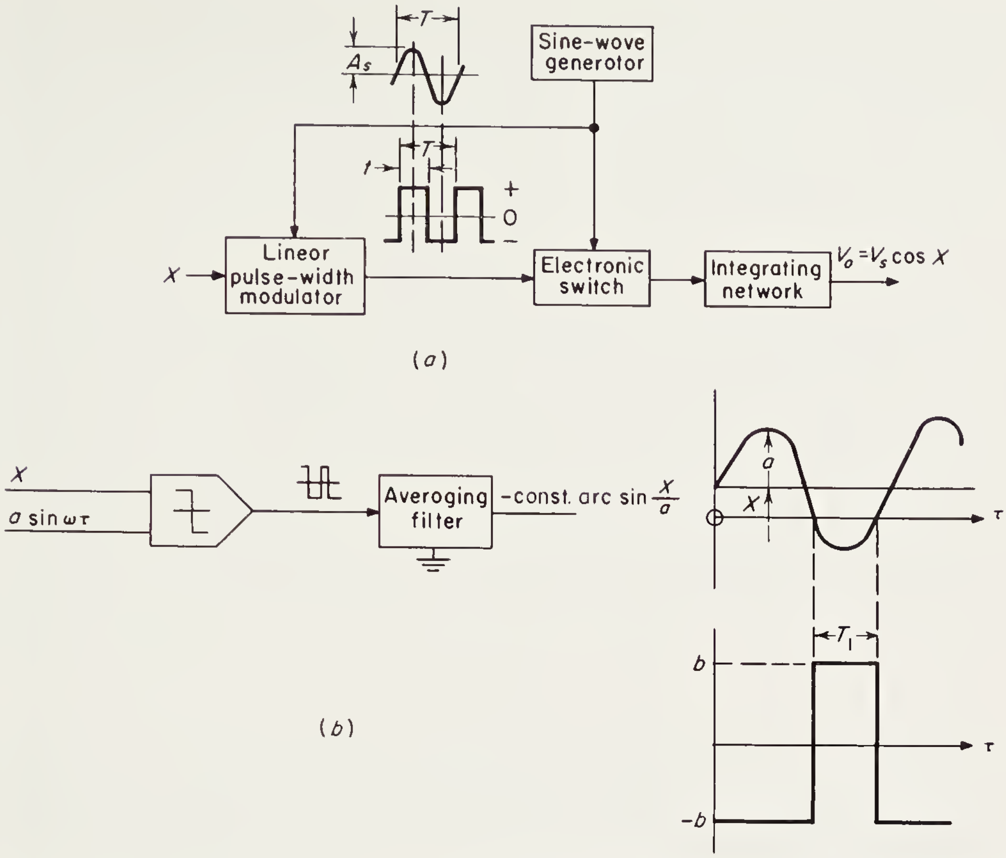


Fig. 6-33. Generation of (*a*) cos X and (*b*) arcsin (X/a) by operations on a periodic sine-wave voltage.

same sine-wave carrier to produce a chopped sinusoidal waveform whose running average is proportional to

$$\omega \int_{-\frac{\pi-2X}{2\omega}}^{\frac{\pi-2X}{2\omega}} \cos \omega\tau \, d\tau = 2 \cos X \tag{6-26}$$

Static accuracy within 0.2 percent of half-scale (±10 volts) is possible if we combine push-pull chopped sinusoids.[129] In the arcsine-function generator of Fig. 6-33*b*, the input voltage X is added to $a \sin \omega\tau$ to produce a biased sinusoid positive and negative for alternate periods $T_1 = (2/\omega) \arccos (X/a)$ and $T_2 = (2/\omega)[\pi - \arccos (X/a)]$; so that the running average of the comparator output is

$$b\,\frac{T_1 - T_2}{T_1 + T_2} = -\frac{2b}{\pi}\left(\frac{\pi}{2} - \arccos \frac{X}{a}\right) = -\frac{2b}{\pi} \arcsin \frac{X}{a} \tag{6-27}$$

where X is measured in radians. Such arcsine generators can have static accuracies within 0.2 percent of half-scale and may be used in the inverse-function loop of Fig. 4-32*a* to yield sin cX.

Each of the circuits of Fig. 6-33 requires an accurate sine-wave oscillator, which can serve several function generators. If some type of phase-shift oscillator is used, we must minimize oscillator amplitude jitter even at the expense of increased distortion (Sec. 4-13), since distortion effects can be corrected with the aid of crude diode correction-function generators in the manner of Fig. 2-5.[133]

6-24. Implicit-computation Schemes. If, as is often the case, a voltage proportional to $\dot{\vartheta}$ is available in the computer, then we can employ the generalized-integrator circuit of Fig. 1-15 to generate $X = 50 \cos \vartheta$ and $Y = 50 \sin \vartheta$ without scale-factor restrictions (Sec. 6-21) as solutions of the differential equations

$$\frac{dX}{d\vartheta} = -Y \qquad \frac{dY}{d\vartheta} = X \tag{6-28}$$

where ϑ is measured in radians (Fig. 6-34*a*). To prevent exponential sine-wave-amplitude changes due to integrator and multiplier phase shift (Sec. 3-24), Howe and Gilbert[119] have devised an ingenious amplitude-correction circuit. Referring to Fig. 6-34*b*, we continuously compute the amplitude-error voltage

$$\epsilon = A\,(\cos^2 \vartheta + \sin^2 \vartheta - 1) = A(X^2 + Y^2 - 1) \qquad (A > 0) \tag{6-29}$$

and minimize ϵ^2 by feeding voltages proportional to

$$\frac{1}{4}\frac{\partial \epsilon^2}{\partial X} = \epsilon X \qquad \frac{1}{4}\frac{\partial \epsilon^2}{\partial Y} = \epsilon Y \tag{6-30}$$

as respective corrections to $-dX/d\tau$ and $-dY/d\tau$ into the integrators producing $X = 50 \cos \vartheta$ and $Y = 50 \sin \vartheta$. Note that the correction voltages are zero whenever the output amplitude is correct. This computer circuit, a special instance of the steepest-descent method discussed

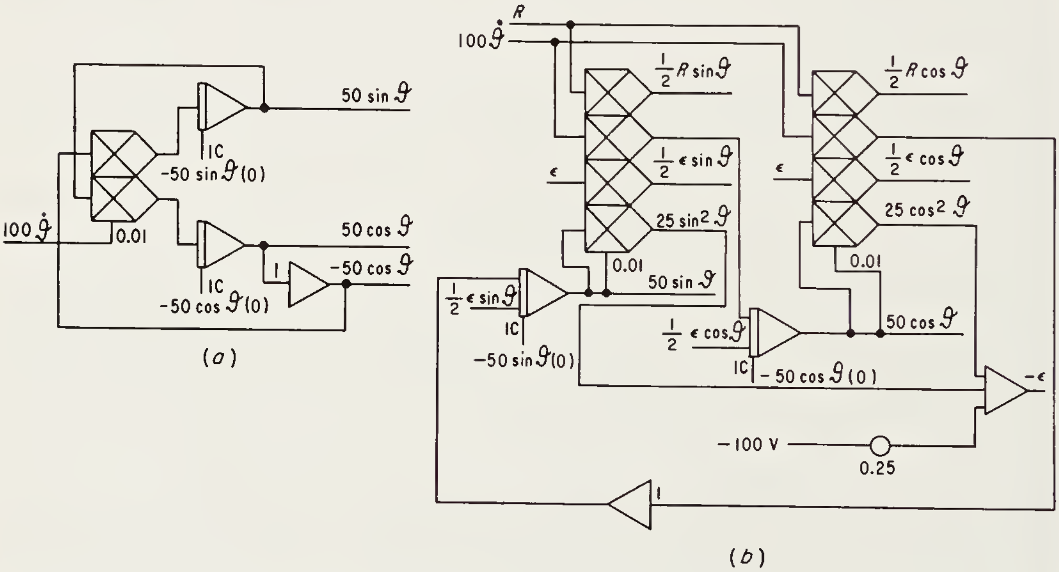


Fig. 6-34. (*a*) Generation of sin ϑ and cos ϑ by implicit computation and (*b*) electronic rate resolver with steepest-descent-type correction of the amplitude error $\cos^2 \vartheta + \sin^2 \vartheta - 1$. (*From H. D. Huskey and G. A. Korn, Computer Handbook, McGraw-Hill, New York,* 1962.) If the 25-volt input to the ϵ amplifier is replaced by a (slowly) variable input $r^2(\tau)/100$, the two integrators will produce the products $r(\tau) \cos \vartheta$, $r(\tau) \sin \vartheta$ with suitable initial conditions, although the response to $r(\tau)$ may be somewhat sluggish (see also Sec. 8-9*b*).

in Sec. 8-9*b*, employs only multipliers and linear computing elements to generate sin ϑ and cos ϑ with about the component accuracy of the multipliers used. Brunner[120] has reduced the relatively large number of electronic multipliers required by minimizing $|\epsilon|$ rather than ϵ^2 (see also Sec. 8-9*b*).

Analogous correction schemes also permit computation of three direction cosines cos α, cos β, cos γ for three-dimensional coordinate transformations; our correction feedback must, in this case, enforce the relation[132]

$$\cos^2\alpha + \cos^2\beta + \cos^2\gamma - 1 = 0 \tag{6-31}$$

Such implicit-computation methods are economical only if the relatively large number of accurate products required can be obtained at reasonable cost. Solid-state time-division multipliers with multiple transistor-switch slave channels (Sec. 6-11) may be suitable for this purpose. Figure 6-34*b* can, of course, also be implemented with two multiproduct servomultipliers.

REFERENCES AND BIBLIOGRAPHY

Computer-servomechanism Design

1. Gilbert, E. O.: The Design of Position and Velocity Servos for Multiplying and Function Generation, *IRETEC*, September 1959.
2. Loveman, B. D.: Computer Servomechanisms and Servo Resolvers, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
3. Harris, M.: Servomultiplier Performance, *Berkeley Engineering*, no. 1, April 1957, Beckman/Berkeley Div., Richmond, Calif.
4. Gibson, J. E., and F. B. Tuteur: *Control System Components*, McGraw-Hill, New York, 1958.
5. Truxal, J. G.: *Control Engineers' Handbook*, McGraw-Hill, New York, 1958.
6. Biernson, G. A.: Comparison of Lead-network, Tachometer, and Damper Stabilization for Electric Servos, *IRE Trans. PGAC*, May 1959.
7. Adise, H.: One Solution to Servomechanism Hunting, *Electron. Inds.*, January 1961.
8. Greenwood, I. A., Jr., J. V. Holdam, Jr., and D. MacRae, Jr.: *Electronic Instruments*, MIT Radiation Laboratory Series, vol. 21, McGraw-Hill, New York, 1948.
9. Gunning, W. F., and A. S. Mengel: *Rept.* RM 236, Rand Corporation, Santa Monica, Calif.
10. Ehlers, H. L., and W. Hochwald: Maintenance Procedures and Test Circuits, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
11. Adise, H. H.: Precision Film Potentiometers, *WESCON Record*, 1960.
12. Nettleton, L. A., and F. E. Dole: Potentiometers, *Rev. Sci. Instr.*, **17**:356 (1946).
13. ———: Reducing Potentiometer Loading Error, *Rev. Sci. Instr.*, **18**: 332 (1947).
14. Cahn, L.: Automatic Compensation of Potentiometer Loading Errors, *Helinews*, Helipot Corporation, South Pasadena, Calif., June 1953.
15. Gilbert, J.: Here's a Shortcut in Compensating Potentiometer Loading Errors, *Control Eng.*, February 1955.
16. ———: Compensating Function Potentiometers for Loading Errors, *Control Eng.*, March 1955.
17. Hofstadter, R.: A Simple Potentiometer Circuit for Production of the Tangent Function, *Rev. Sci. Instr.*, **17**:298 (1946).

18. Levenstein, H.: Generating Non-linear Functions with Linear Potentiometers, *Electron. Inds.*, October 1953.
19. Seay, P. A.: An Accurate Tangent Computing Circuit, *REAC Newsletter*, August 1954 (Reeves Instrument Corp., New York).
20. Bolie, V. W.: Synthesis of Function Generators by Continued-fraction Theory, *Electron. Inds.*, March 1956.
21. Lazarus, H.: Cosine Computing Circuit, U.S. Patent 2,864,924, September 1958 (assigned to Reeves Instrument Corp.).

General

22. Connelly, M. E.: Computers for Aircraft Simulation, *MIT Electron. Sys. Lab., Rept.* 7591-R2, Cambridge, Mass., 1959.
23. Morrill, C. D.: Electronic Multiplier and Related Topics, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.

Miscellaneous Quarter-square Multipliers

24. Soltes, A. S.: Wide-band Square-law Circuit Element, *IRETEC*, April 1955.
25. Miller, I. A., A. S. Soltes, and R. E. Scott: Wide-band Analog Function Multiplier, *Electronics*, **28** 1955.
26. Fitsner, L. N.: A Multiplication Block Using Thyrites, *Priborostroenie*, no. 4, 1956.
27. Maslov, A. A.: A Multiplier-divider Based on Thyrites, *Automatika i Telemekhanika*, **18** April 1957.
28. Kovach, L. D., and W. Comley: A New Solid-state Nonlinear Analog Component, *IRETEC*, December 1960.
29. Schreier, D., and H. Winkler: Design of Nonlinear Characteristics Using Varistors, *Proc. AICA 3d Intern. Conf.*, Opatija, Yugoslavia, September 1961, Presses Académiques Européennes, Brussels.
30. Bloom, H. M.: Circuit Parameters for Optimum Quadratron Performance, *Harry Diamond Lab. Tech. Mem.* 66–15, Washington, D.C., 1966.
31. Kurokawa, K., and A. Yagi: Multiplier Using MOSTs as Square Element, *Min. Intern. Trade Ind. Electrotech. Lab. Tech. Note*, Tokyo, 1968.

Diode Quarter-square Multipliers

32. Fisher, M. E.: A Wide-band Analog Multiplier Using Crystal Diodes, *Electron. Eng.*, **29**:358 (1957).
33. Fisher, M. E.: The Optimum Design of Quarter-square Multipliers with Segmented Characteristics, *J. Sci. Instr.*, **34** August 1957.
34. Sternick, L.: Technical Data Sheet on M-400-1 REAC Diode Electronic Multiplier, *Rept.* TD5567, Reeves Instrument Corp., Garden City, N.Y., 1956.
35. McCoy, R. D., and H. F. Meissinger: U.S. Patent 3,031,143, assigned to Reeves Instrument Corp., Apr. 24, 1962 (filed Jan. 7, 1955).
36. Whigham, R. H.: A Fast 10-volt Diode Quarter-square Multiplier, *Simulation*, August 1965.
37. MacKay, D. M., and M. E. Fisher: *Analog Computing at Ultra-high Speed*, Wiley, New York, 1962.
38. Ream, N.: Approximation Errors in Diode Function Generators, *J. Electron. Control*, **7**:83 (1959).
39. Ritchie, C. C., and R. W. Young: The Design of Biased Diode Function Generators, *Electron. Eng.*, **31**:347 (1959).
40. Deiters, R. M.: Optimum Design of a Diode Squarer, *IEEETEC*, June 1965.
41. Giloi, W., and R. Fritz: Computer-aided Design of Quarter-square Multipliers, *Simulation*, September 1968.

Folding Techniques for Squaring Circuits

42. Netter, Z.: Folding Technique Extends Function-generator Range, *Electron. Design*, May 10, 1963.
43. Nathan, A.: Interpolating Multipliers and Related Interpolators, *Proc. IEEE*, November 1963.

44. Nathan, A.: Cascade Squarer, *Proc. IEEE*, April 1965.
45. ——— and G. Gafni: Design and Performance of Nonlinear Interpolating Squarers, *Proc. IEEE*, July 1965.
46. ——— and J. Molcho: Improved Voltage Selector and Cascade Multiplier Circuits, *IEEETEC*, April 1968.
47. Howe, R. M.: Some Techniques for Accuracy Improvement in Analog Computation, *Simulation*, October 1967.

Triangle-averaging Multipliers

48. Meyer, R. A., and H. B. Davis: Triangular-wave Analog Multiplier, *Electronics*, August 1956.
49. Pfeiffer, P. E.: A Four-quadrant Multiplier, *IRETEC*, June 1959.
50. G. A. Philbrick Researches, Boston, Mass., *Specification Sheets for MU/DV and K5-M Multipliers.*
51. Hartmann, J., G. A. Korn, and R. L. Maybach: Low-cost Triangle-integration Multipliers for Analog Computers, *Ann. AICA*, October 1961.
52. Alliston, W. H.: A Four-quadrant Multiplier, *Simulation*, January 1965.

Logarithmic Multipliers

(See the Bibliography for Chap. 4)

(Transistor-) Transconductance Multipliers

53. Morton, R. R.: A Simple D-c to 10 Mhz Analog Multiplier, *J. Sci. Instr.*, **43**:165 (1966).
54. Haase, W. C.: "A Novel Analog Multiplier", M.S. Thesis, Massachusetts Institute of Technology, 1967; see also *Simulation*, 1970.
55. Gilbert, B.: The Current-gain Cell, *EDN*, June and July 1968.
56. ———: A Precise Four-quadrant Multiplier with Subnanosecond Response, *IEEE J. Solid-state Cts.*, December 1968.
57. Free, M. G.: "An Integrated Linear-transconductance Analog Multiplier", M.S. Thesis, The University of Arizona, 1969; *Simulation*, November 1969.

Frequency- and Time-sharing Feedback Multipliers

58. RCA Laboratories Division, unpublished report on Project Typhoon, 1949.
59. Chance, B., et al.: *Waveforms*, vol. 19, sec. 19.5, MIT Radiation Laboratory Series, McGraw-Hill, New York, 1949.

Heat-transfer and Photoresistor Multipliers

60. Savet, P. H.: Heat-transfer Computing Elements, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962 (includes bibliography).
61. Connelly, M. E.: An Analog Photoresistive Multiplier, *Simulation*, November 1966.
62. The Dual Photocell, *Application Note* 924, Hewlett-Packard Corporation, 1967.
63. Soxman, E., et al.: *Development of Thin-film CdS Arrays for an Analog Multiplier*, Sigmatron, Inc., Goleta, Calif., 1969.

Early Multipliers Employing Analog/Digital Conversion

64. Goldberg, E. A.: Step Multiplier in Guided Missile Computer, *Electronics*, **24**:120 (1951).
65. Schmid, H.: Transistor Analog Computing Elements, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
66. Maslov, A. A.: Survey and Classification of Multiplying Devices, *Automatika i Telemekhanika*, **21** October 1960.

Hall Effect and Magnetoresistance Multipliers

67. Lofgren, L.: Analog Multiplier Based on the Hall Effect, *J. Appl. Phys.*, **29**: 153 (February 1958).

68. Oxeni, J.: Hall-effect Multipliers, *Nachr. Tech. Z.*, May 1958.
69. Scanga W. A., A. R. Hilbinger, and C. M. Barrack: Hall-effect Multipliers, *Electronics*, July 15, 1960.
70. Cohen, E.: A Hall-effect Multiplier for Use at Radio Frequencies, *Electron. Eng.*, September 1960.
71. Fuchs, H., and D. G. Flockhart: A Hall-effect Analog Multiplier, *Electron. Eng.*, November 1960.
72. Kovatch, G., and W. E. Meserve: The Hall-effect Analog Multiplier, *IRETEC*, September 1961.
73. Greiner, R. A.: Feedback Improves Hall-effect Multipliers, *Electronics*, Aug. 25, 1961.
74. Schmid, H.: Transistor Analog Computing Elements, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
75. Hunt, J. M.: Application of the Magnetoresistance Effect to Analog Multiplication, *Stanford Electron. Lab. Tech. Rept.* 1504-1, Stanford, Calif., May 25, 1960; also *Trans. Natl. Electron. Conf.*, 1960.
76. Kataoka, S.: Magnetoresistance Multiplier with Higher Gain, *Proc. IRE*, February 1962.

Coincidence Multipliers

77. Chance, B., et al.: *Waveforms*, vol. 19, Sec. 19.5, MIT Radiation Laboratory Series, McGraw-Hill, New York, 1949.
78. Siljak, D., and R. Petrovic: Multiplication by Means of Coincidences, *Proc. 3d AICA Intern. Conf.*, Opatija, Yugoslavia, September 1961; Presses Académiques Européennes, Brussels.

AM/FM Multipliers

79. Price, R.: An FM-AM Multiplier of High Accuracy and Wide Range, *MIT Res. Lab. Electron. Tech. Rept.* 213, Cambridge, Mass., October 1951.
80. Sommerville, M. J.: An Electronic Multiplier, *Electron. Eng.*, February 1952. McCool, W. A.: An AM-FM Electronic Analog Multiplier, *Proc. IRE*, **41**:1470 (October 1952).
81. AM-FM Demodulator Used in Multiplier System, *Electron. Design News*, July 1959.

Other Modulation Schemes

82. Keister, G. L.: Transistor Magnetic Analog Multiplier, *Electronics*, October 1953; see also Refs. 2, 7, and 21; and U.S. Patent 2,784,909 (1957), assigned to General Electric Co.
83. Wilson, J. P.: A Simple High-speed Analog Multiplier, *Electron. Eng.*, January 1967.
84. Bohn, E. V.: A Pulse-position-modulation Analog Computer, *IRETEC*, June 1960.
85. Paul, R. J. A.: Hybrid Methods for Function Generation, *Proc. 3d AICA Conf.*, Opatija, Yugoslavia, 1961, Presses Académiques Européennes, Brussels, 1962; see also *Control Eng.*, 1962.
86. Schaefer, R. A.: Pulse Modulator for Accurate D-c Amplification, *IRE Trans. PGI*, September 1962.
87. ———: New Pulse Modulation Method, *Electronics*, Oct. 12, 1962.
88. Crayford, R. M.: APD Coding and Logic, *Instr. Control Systems*, January 1962.
89. Seegmiller, W. R.: Accurate Analog Computation with Pulse-time Modulation, *Electronics*, Mar. 30, 1962.
90. Digital-Analog Pulse-time Circuits, Staff Report, *Computer Design*, June 1963.
91. Schmid, H.: "A Hybrid Operational Computing System," M.S. Thesis, Syracuse University, 1963; see also *IEEETEC*, October 1963.
92. ———: A Repetitive Analog Computing Technique with a Digital Integrator, *Proc. Intern. Aerospace Conf.*, Phoenix, Ariz., 1964.
93. ——— and B. Gringle: A Precision Pulse-width Modulator, *Electronics*, May 12, 1964.

Time-division Multipliers

94. Sack, H. S.: *NDRC Rept.* 14-435, Cornell University, Ithaca, N.Y., 1944; see also Cornell thesis of A. C. Beer and H. W. Boehmer.
95. Greenwood, I. A., J. V. Holdam, Jr., and D. MacRae, Jr.: *Electronic Instruments*, MIT Radiation Laboratory Series, vol. 21, McGraw-Hill, New York, 1948.
96. Millman, J., and H. Taub: *Pulse and Digital Circuits*, McGraw-Hill, New York, 1956.
97. Goldberg, E. A.: A High-accuracy Time-division Multiplier, *RCA Rev.*, **23**:265 (September 1952).
98. Morrill, C. D., and R. V. Baum: A Stabilized Electronic Multiplier, *IRETEC*, December 1952.
99. Walters, L. G.: "A Study of the Series-motor Relay Servomechanism," Ph.D. Thesis, University of California, Los Angeles, 1951.
100. Korn, G. A., and T. M. Korn: Relay Time-division Multiplier, *Rev. Sci. Instr.*, **25**:977 (October 1954).
101. Sternberg, S.: An Accurate Electronic Multiplier, *RCA Rev.*, **16**:618 (December 1955).
102. Lilamand, M.: A Time-division Multiplier, *IRETEC*, January 1956.
103. Schmid, H.: A Transistorized Four-quadrant Time-division Multiplier with an Accuracy of 0.1 percent, *IRETEC*, March 1958.
104. Gleghorn, P.: An Analog Electronic Multiplier Using Transistors as Square-wave Modulators, *Proc. IRE*, **107(B)**:94 (March 1960).
105. Schneider, W.: Ein transistorisierter Time-division-Multiplikator hoher Genauigkeit, *Telefunken Z.*, September 1960.
106. Kettel, E., and W. Schneider: An Accurate Analog Multiplier and Divider, *IRETEC*, June 1961.
107. Bode, H. W.: *Network Analysis and Feedback-amplifier Design*, Van Nostrand, Princeton, N.J., 1945; see also Ref. 2.

Sampling Multipliers and Resolvers

108. Broomal, J., and L. Riebman: A Sampling Analog Computer, *Proc. IRE*, May 1952.
109. Miura, T.: Japanese patent issued in 1958.
110. Andrews, J.: The Operational Amplifier with DYSTAC as a General-purpose Nonlinear Element, *Application Data Sheet* 0005, Computer Systems, Inc., Fort Washington, Pa., 1960.

Electron-beam Multipliers

111. MacNee, A. B.: An Electronic Differential Analyzer, *Proc. IRE*, **37**:1315 (1949).
112. Schmidt, W.: Die Hyperbelröhre, eine Elektronenstrahlrohre zum multiplizieren in Analogie-rechengeräten, *Z. Angew. Phys.*, no. 2, 1956.
113. Gundlach, F. W.: A New Electron-beam Multiplier with an Electrostatic Hyperbolic Field, *Proc. AICA 1st Intern. Conf.*, Brussels, 1955, Presses Académiques Européennes, Brussels.
114. Angelo, E. I.: An Electron-beam Tube for Analog Multiplication, *Rev. Sci. Instr.*, **25** (3) (March 1954).

Precision Multiplication Using Multiple Channels

115. Fitsner, L. M.: A Precision Multiplier, *Automatika i Telemekhanika*, **20** (January 1959).
116. Maslov, A. A.: A Three-channel Multiplier with Frequency Division of the Signals, *Automatika i Telemekhanika*, **21** (December 1960).
117. Schmid, H.: Combined Analog-Digital Computing Elements, *Proc. Western Joint Computer Conf.*, 1961.
118. Stojanovic, B., and D. Milicevic: Time-division System for Multiplication, *Proc. IMEKO*, Versailles, France, 1970.

Electronic Resolvers and Inverse Resolvers

119. Howe, R. M., and E. G. Gilbert: Trigonometric Resolution in Analog Computers by Means of Multiplier Elements, *IRETEC*, June 1957.
120. Brunner, W.: Steepest Ascent, *PCC Rept.* 151, Electronic Associates, Inc., Princeton, N.J., 1959.
121. Korn, G. A.: The Impact of Hybrid Analog-Digital Techniques on the Analog-computer Art, *Proc. IRE*, May 1962.
122. Raymond, F. H.: Methode d'intégration fractionnée, *Ann. AICA*, July 1960.
123. Pinckney, R. P., and S. Giser: The Sine-Cosine Operator, *Rept.* R-73, MIT Instrumentation Laboratory, Cambridge, Mass., 1954.
124. Morrill, C. D.: Electronic Multipliers and Related Topics, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
125. Thaulow, P.: Electronic Resolver System, *Memo. Rept.*, Beckman/Berkeley Division, Richmond, Calif., Dec. 4, 1959; see also Beckman/Berkeley specification sheets for Models 1154 sine/cosine generator.
126. Sine-Cos Generator Model 8.054: *Product Information Release* 6079, Electronic Associates, Inc., Long Branch, N.J., 1960.
127. Grado, G. R.: A Solution to the Euler-angle Transformations, *IRETEC*, September 1960.
128. Kindle, W. K.: High-accuracy, High-speed Electronic Resolver, *Eng. Appl. Release* 4-1, Electronic Associates, Inc., Long Branch, N.J., 1961.
129. Hartmann, J., G. A. Korn, and R. L. Maybach: Low-cost Triangle-integration Multipliers for Analog Computers, *Ann. AICA*, October 1961.
130. Schmid, H.: A Transistorized, All-electronic Cosine/Sine Function Generator, *WESCON Record*, 1958.
131. ———: Function Generator for Sines and Cosines, *Electronics*, Jan. 23, 1959.
132. ———: Transistor Analog Computing Elements, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
133. Maybach, R. L.: "A Sine-Cosine Generator for Use in Analog Computers", M.S. Thesis, Electrical Engineering Department, University of Arizona, 1961.
134. Vichnevetsky, R.: *EAI Rept. ECC* 59, Electronic Associates, Inc., Brussels, July 1962.
135. Anderson, T., and C. J. Sleehan: Continuous Electronic Polar Resolver, *Instr. Control Systems*, February 1963.
136. Markson, E. E., and J. L. Stricker: The Integrated Mission Simulator, *Simulation*, February 1964.
137. Mitchell, E. E.: Note on Electronic Resolvers, *Simulation*, July 1964.
138. Schmid, H.: Integrated Circuits Replace the Electromechanical Resolver, *Electronics*, January 1966.
139. Kley, A., and E. Heim: Ein elektronischer Koordinaten-Umwandler, *Telefunken Z.*, January 1966.

Miscellaneous

140. Deeley, E. M., et al.: Two New Tubes for High-speed Analog Multiplication, *Proc. IEE*, July 1967.
141. FETs as Voltage-controlled Resistors, *Application Note* Nov. 15, 1966, Siliconix, Inc., Sunnyvale, Calif.
142. Ginzburg, S. A.: Logical Method of Synthesizing Function Generators, *Proc. 4th AICA Conf.*, Lausanne, Switzerland, 1967.
143. Korn, G. A., and T. M. Korn: *Electronic Analog Computers*, McGraw-Hill, New York, 1956.

CHAPTER 7

COMPUTER SYSTEMS AND INTERFACE DESIGN

SPECIAL-PURPOSE SYSTEMS

7-1. Special-purpose Computer Systems and Subsystems. An overwhelming majority of analog and hybrid-computer applications employ the circuits and techniques described in Chaps. 1 to 6 in *special-purpose computer systems* rather than in the better known general-purpose analog/hybrid computers. Operational amplifiers, analog multipliers, function generators, comparators, and switches, fabricated as integrated circuits or epoxy-cast discrete-component assemblies, are combined with or without digital-computer interface logic. Such components appear on etched-circuit cards mounted in simple card cages, special environment-proof enclosures, or component ovens. ±10-volt analog computing elements with ±15-volt power supplies are preferred for such applications and can be purchased for "commercial" temperature ranges between 0 and 50 deg/C or for military/space temperature ranges (−50 to +150 deg/C). Familiarity with such analog/hybrid computer-system components and their applications can be invaluable to designers of instrumentation and control systems, because a $300 handful of integrated-circuit amplifiers, diodes, resistance networks, and logic may be miraculously tranformed into a $1,500 subsystem card when purchased from an instrument manufacturer.

GENERAL-PURPOSE ANALOG/HYBRID COMPUTERS

7-2. Typical Installations and General Design Principles. (a) Small Tabletop Computers. Small analog computers incorporating between 6 and 24 ±10-volt amplifiers, plus coefficient-setting potentiometers and

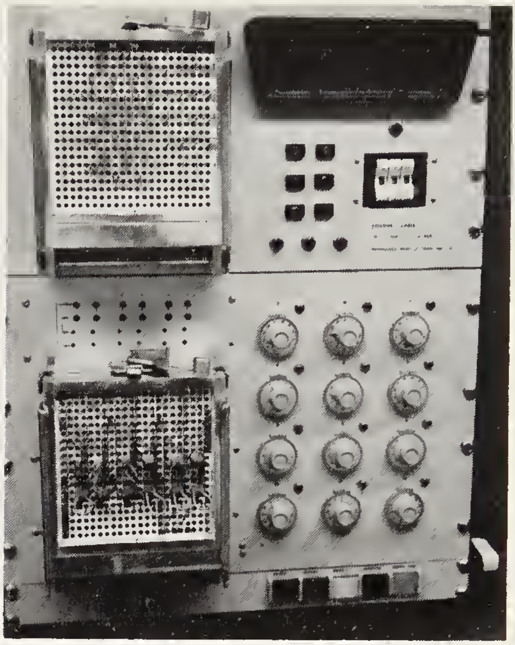


Fig. 7-1*a*. The University of Arizona's APE II has 12 operational amplifiers (with 20-MHz unity-gain bandwidth), 2 multipliers, 2 comparators, and assorted diodes and switches mounted on cards plugged directly into the analog (bottom) patchbay. The digital (top) patchbay has a digital clock, 1 preset and 8 free decimal counters, 10 OR/NOR gates, 5 free flip-flops, plus 20 shift-register flip-flops with 2 XOR circuits for sequence generation, pushbuttons, indicator lights, and a counter readout. Prewired "experiment boxes" committing the amplifiers to specific student experiments can replace the analog patchboard. The machine is described in detail in Refs. 10 to 13.

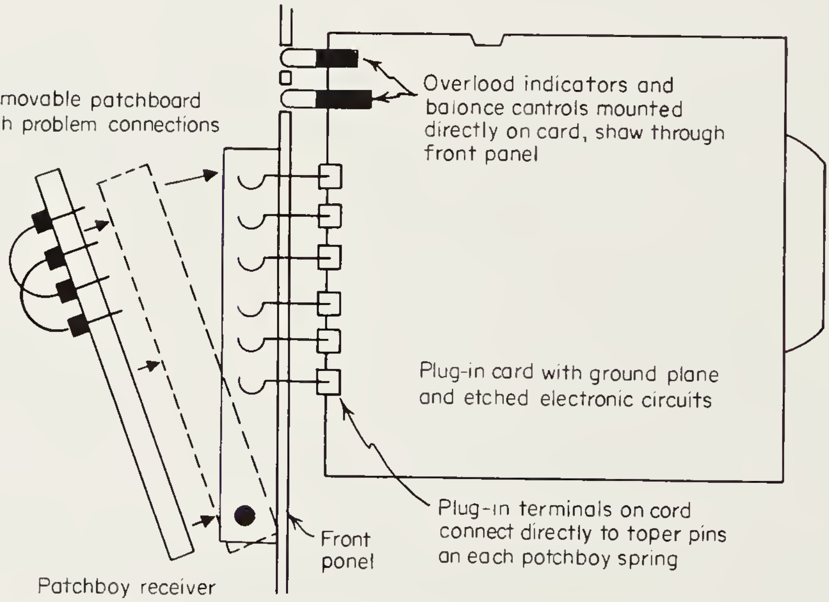

Removable patchboard with problem connections
Overload indicators and balance controls mounted directly on card, show through front panel
Plug-in card with ground plane and etched electronic circuits
Plug-in terminals on card connect directly to taper pins on each patchbay spring
Front panel
Patchbay receiver

Fig. 7-1*b*. This figure illustrates the simplicity of APE II construction. Unused patchbay points (with their patchbay springs broken off) terminate power-supply and mode-control connections, so that no wiring to the circuit cards is needed. (*University of Arizona; from Ref.* 10.)

some multipliers and function generators are widely used in educational institutions to teach ordinary differential equations and their applications to control systems, chemical reactions, and other dynamical systems. Modern computers of this type have electronic mode-control switching and permit repetitive operation and simple iteration. More importantly, modern tabletop hybrid computers[10] can incorporate a great deal of inexpensive patchable integrated-circuit logic, viz., clocks, gates, flip-flops, counters, and shift registers, and hybrid computing elements such as analog comparators, track-hold circuits, and simple A/D and D/A converters. Analog computing elements used in such machines are the same rugged and inexpensive ±10-volt modules with ±15-volt power supplies employed in special-purpose hybrid computing equipment (Sec. 7-1). Such small hybrid computers are ideal tools for *breadboarding* as well as for simulation of all kinds of computer systems used in control, instrumentation, statistical data processing, and digital-computer interfacing. Figure 7-1 illustrates a modern tabletop computer of this type used in various electrical-engineering courses at the University of Arizona, and Table 7-1 lists some of the student experiments possible with this small hybrid computer.

Table 7-1. Examples of APE II Experiments*

A. Simulation techniques
1. Aircraft/autopilot simulation
2. Simulation of a nuclear reactor
3. Bouncing-ball simulation using analog memory
4. Simulation of a sampled-data control system; solution of difference equations
5. Solution of a two-point boundary-value problem by iterative optimization
6. Simple model-matching experiments
B. Hybrid-computer hardware techniques
1. Study of fast diode circuits and precision limiters
2. Study of a simple analog-to-digital converter
3. Accurate signal generators and pulse-width modulators
4. Study of track-hold operation and data reconstruction
5. Simulation of combined analog-digital computation; error compensation
C. Statistical measurements and communication/detection system simulation
1. Measurement of continuous and sampled-data averages
2. Measurement of correlation functions and delay error
3. Polarity-coincidence correlation
4. Measurement of impulse response by input-output correlation
5. Measurement (estimation) of probability and probability density
6. Demonstration of modulation waveforms with and without noise
7. Simulation of a superheterodyne receiver
8. Simulation of an FM communication system
9. Simulation of a phase-lock loop
10. Simulation of matched-filter detection using pseudorandom signals (shift-register sequences)

* See also Ref. 10.

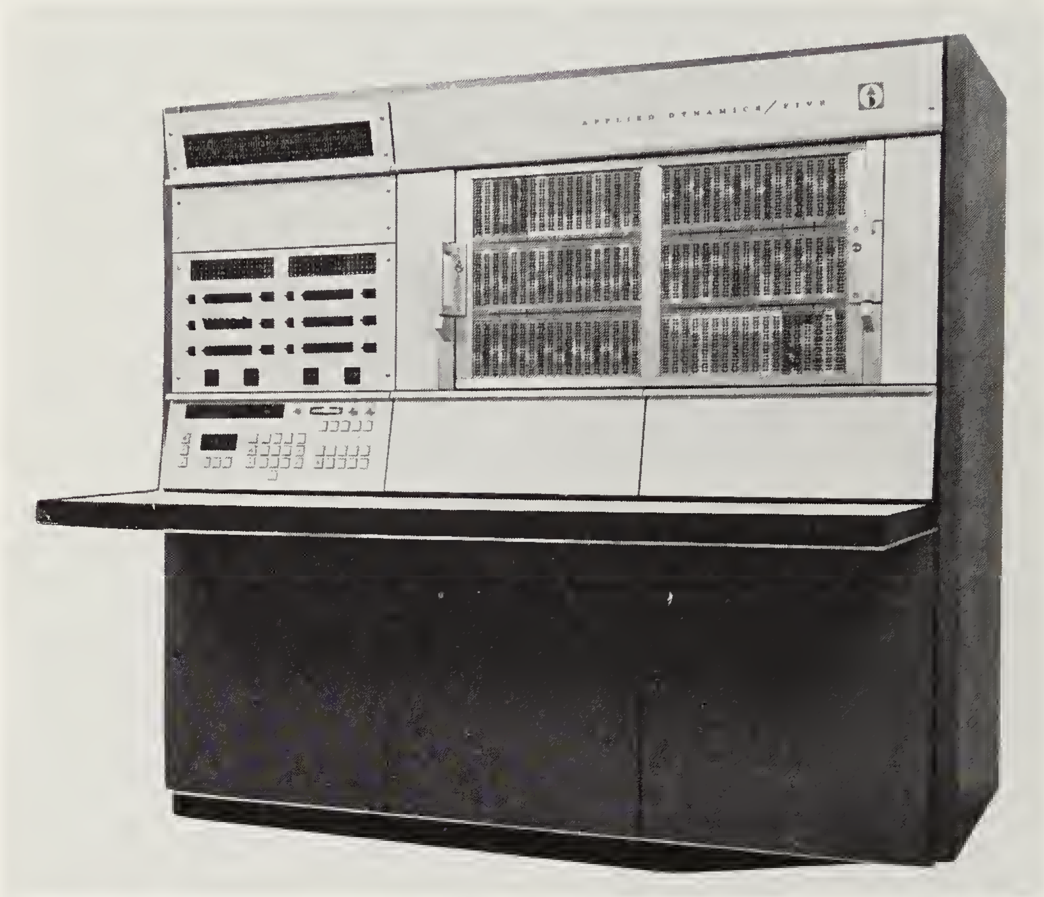


Fig. 7-2*a*. The Applied Dynamics AD/FIVE medium-sized ±10-volt hybrid computer has, when fully expanded, 240 operational amplifiers, of which 36 are integrator/summers and about 80 are committed to some 50 function generators and quarter-square multipliers. 0.01 percent resistance networks are used, and linear-circuit errors are within 0.1 percent up to about 1 KHz. Patchbay-spring modules of the unshielded patchbay have rear card receptacles for both amplifiers and patchable logic. There are up to 60 gates, 48 flip-flops, and 8 two-decade BCD counters, with 16 analog comparators and up to 48 electronic switches. Coefficient-setting potentiometers would be mounted in the front panels below the patchbay but have been replaced with digital coefficient-setting units in the machine shown. The panel above the patchbay holds diode function generators. The control panel on the left has overload lights, logic-state indicators for the patched digital logic, counter controls, and a keyboard addressing system and digital voltmeter for setting coefficients and checking analog component output voltages. A hybrid-computer interface is available as an integral component of the AD/FIVE, which can also be supplied with a Digital Equipment Corporation PDP-11 16-bit minicomputer built into an expanded console. Several AD/FIVE consoles can be slaved together. (*Applied Dynamics, Inc., Ann Arbor, Mich.*)

Small analog computers with or without storable patchboards and digital logic are also very useful in the laboratory for making up special signal generators (Secs. 4-12 and 4-13), signal-conditioning circuits, and test systems.

(b) Medium-size and Large Installations. Larger analog/hybrid computers range from medium-size installations (between 50 and 150

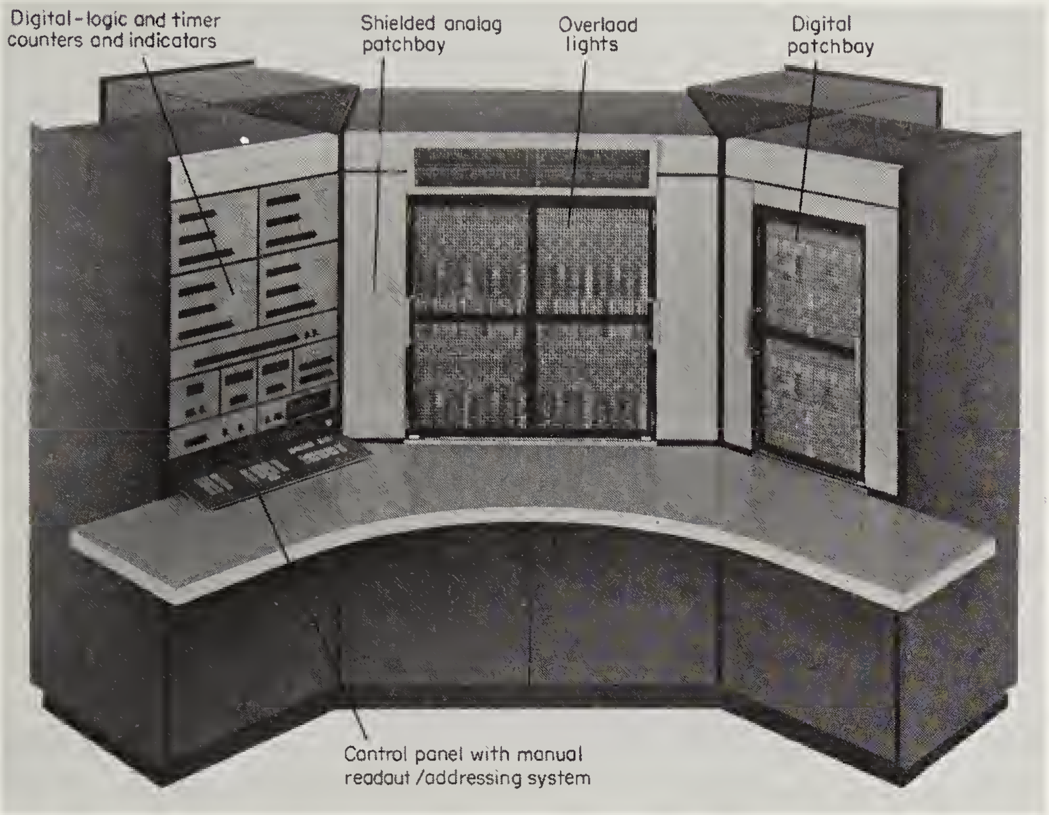

Digital-logic and timer counters and indicators
Shielded analog patchbay
Overload lights
Digital patchbay
Control panel with manual readout /addressing system

Fig. 7-2*b*. Applied Dynamics AD/FOUR is a large ±100-volt machine capable of expansion to several hundred amplifiers with multiple consoles. Interesting features of this machine are summing-junction patching of coefficient-setting units, which permits DCU gains up to 6.5432 and potentiometer gains to 1.6358; optional dielectric-absorption compensation (Fig. 3-18*c*); and optional automatic patching for one patchbay quadrant. (*Applied Dynamics, Inc., Ann Arbor, Mich.*)

amplifiers) to large computers, which can employ multiple consoles to accommodate as many as 1,000 amplifiers and appropriate nonlinear equipment. Both ±10- and ±100-volt computers are used; the larger American-made machines usually employ ±100 volts. High-quality ±10-volt analog computers can match the accuracy of ±100-volt machines,[9] although the smaller dynamic range tends to make nonlinear operations less accurate or more expensive; ±10-volt operation reduces heat dissipation and makes it easier to mount computing elements directly behind the computer patchbay (Fig. 7-1*b*); ±10-volt operation also favors computer bandwidth, both because faster amplifiers are available, and because the lower power dissipation permits one to use lower computing impedances.

System design begins with selection of the voltage scale, current output, and bandwidth required of the d-c amplifiers. Experience shows that computers will be expanding as users begin to realize their possibilities, so that ease of expansion is an important design consideration. Increasing installation size does increase current demand on individual amplifiers, but only moderately so; ±30- to ±50-mA amplifiers are generally sufficient. Figures 7-2 and 7-3 show typical computer consoles.

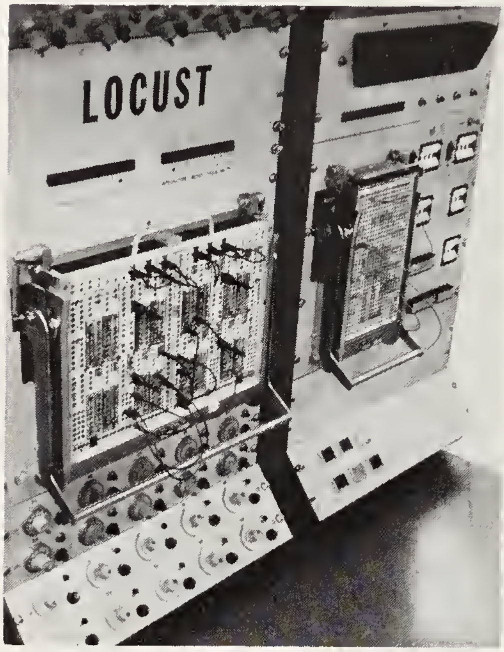


Fig. 7-3. Ten times faster than commercially built analog computers, the 40-amplifier LOCUST can make up to 4,000 differential-equation-solving runs per second under control of a small digital computer. LOCUST amplifiers have 30-MHz 0-db bandwidth, with full ±10-volt 30-mA output to 5 MHz; integrators switch into COMPUTE in 40 nsec and within 10 nsec of one another. Phase-inverter error is 0.1 percent at 10 KHz, using 1K resistors. Analog and digital circuits plug directly into the rear of the patchbay receivers. Emitter-coupled integrated-circuit logic minimizes digital noise in the wideband analog circuits. (*University of Arizona, Ref.* 14; *see also Figs.* 3-35, 4-17, 4-35, 5-29, 5-35, *and* 6-4.)

(c) Power Supplies. Analog computers must incorporate positive and negative *computing-element power supplies* (Table 7-2) and *reference power supplies* (±10 volts or ±100 volts). With all reference voltages slaved to a single system reference, reference-voltage accuracy will affect only nonlinear operations and external instruments but should, in general, at least match computing-element accuracies at all working-load currents and temperatures.

7-3. Logic and Control Systems. (a) Mode Control and Patched Digital Logic. While earlier generations of electronic analog computers employed relays for integrator mode-control switching,[2] the modern control circuits of Figs. 1-10, 5-34, and 5-35 all use electronic switches to switch into COMPUTE or HOLD when appropriate digital-logic voltages are high (digital 1; see Sec. 2-2 and, specifically, the footnote to Sec. 2-2*b*). Three-mode (RESET, COMPUTE, HOLD) integrators require a second logic input to go into the HOLD mode. All state-variable integrators for a set of differential equations are normally controlled by *master* RESET *and* HOLD *logic buses* which can, if desired, control the entire machine. It pays, however, to patch the integrator-control logic in a *digital patchbay*, which also terminates other switch logic, clocks, comparator outputs, and free logic ("logic box"). Patched logic and/or a digital-computer control register (Sec. 7-8) also brought out to the

Table 7-2. Specifications for Typical Analog-computer Power Supplies
Note that actual requirements vary widely

Voltage change for a 5-percent line-voltage change	0.5 to 0.005 percent
Voltage change for a load-current change of 50 percent of rated load, and recovery time to 99.9 percent of steady state	0.5 to 0.005 percent; 250 to 2 μsec
24-hr drift, hum, and noise	0.5 to 10 mV
Source impedance	0.1 to 0.01 ohm (1 to 10 KHz); shunt capacitance and decoupling at higher frequencies

digital patchbay can then switch individual integrators, e.g., as track-hold circuits (Secs. 2-3, 2-4, and 8-14). Figure 7-4 shows a practical digital-patchbay layout for this purpose.

In the POTSET mode, relays switch summing-resistor junctions away from each amplifier and ground them to permit coefficient-potentiometer setting without amplifier overloads. POTSET relays may, instead, decrease summing-amplifier feedback resistances. An even better scheme[14] is to provide all amplifiers with fast limiters (Fig. 4-8*d*) which keep summing junctions at zero voltage even during voltage overloads.

(b) Timing Circuits. Figure 7-6 illustrates a simple and convenient system for timing analog-computer runs, sample readouts, and miscellaneous time intervals in analog or hybrid computation. A 4-MHz crystal-oscillator output is counted down to provide master *clock pulses* at pulse rates of 2 MHz, 1 MHz, and 500, 250, 100, 50, and 10 KHz. The clock rate to be used for the analog-computer master clock ("clock" in Fig. 7-6) is selected by a *time-scale switch* or through logic controlled by the digital computer. *Clock counters* preset manually or by the digital computer can now accurately establish the duration of COMPUTE and RESET periods and of various sampling times. It is easy to produce either a single run or repetitive operation; a typical repetition period is given either 1,000 or 10,000 clock intervals. Either the digital computer

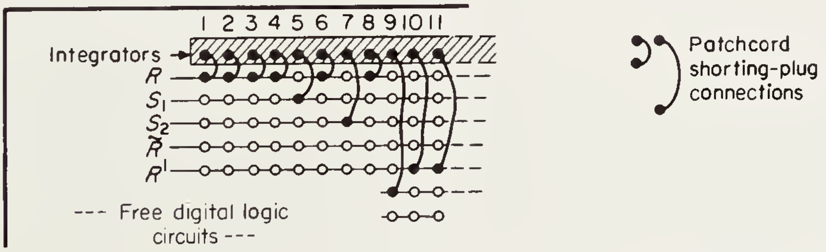

1 2 3 4 5 6 7 8 9 10 11
Integrators
R
S1
S2
R̄
R'
--- Free digital logic circuits ---
Patchcord shorting-plug connections

Fig. 7-4. Digital-patchbay layout which permits logic control of individual integrators, with the possibility of simple shorting-link patching to the frequently used *R*-bus.

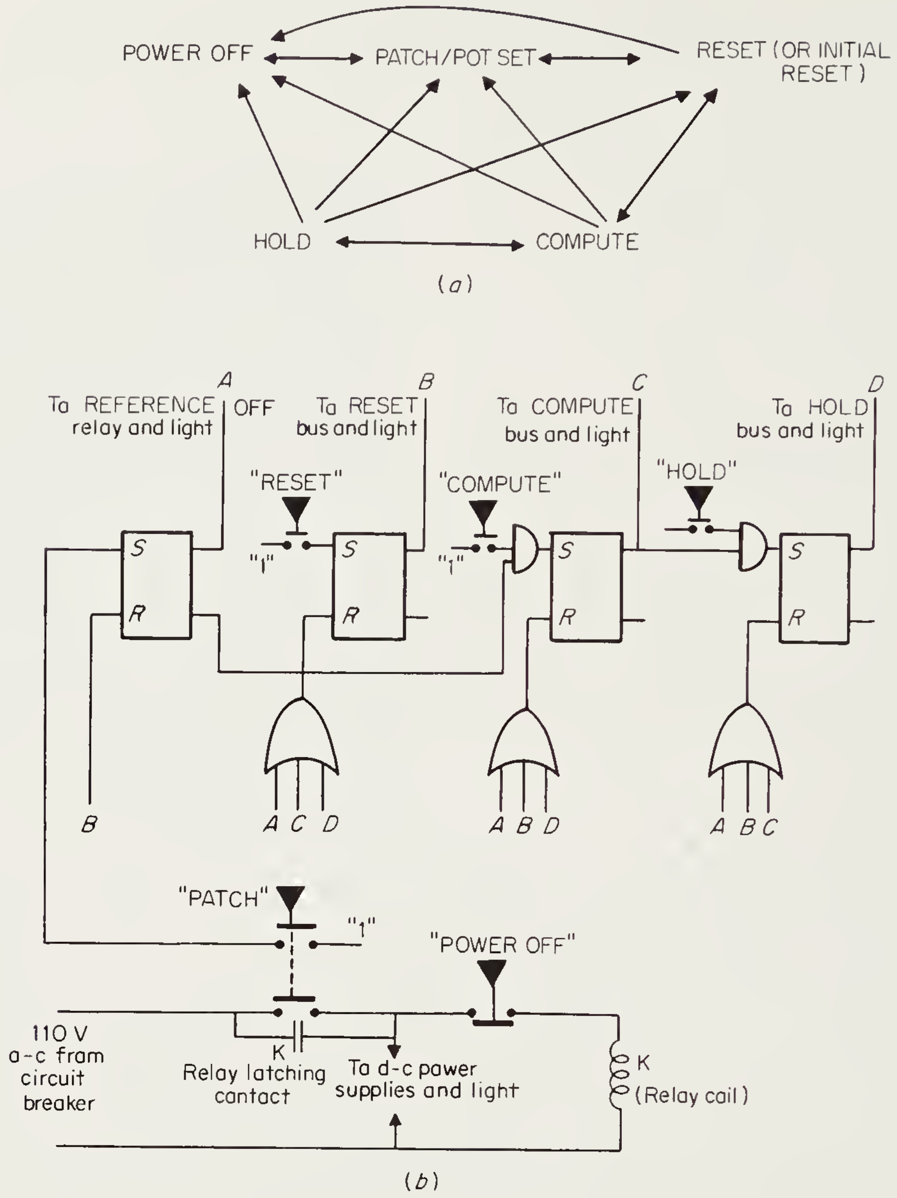


Fig. 7-5. Permissible state sequences for a simple analog computer (*a*) and a control-circuit implementation illustrating both latching-relay and set/reset (*R-S*) flip-flop logic (*b*). In the PATCH/POTSET state, reference power is removed from the patchbay but not from the potentiometer calibration circuits, to permit patching and coefficient setting. Pot-set relays are obsolete with modern fast overload limiters (Sec. 4-5).

or patched digital logic can readily change readout sampling times from run to run (Fig. 7-6*c*). In iterative analog computation, repetitive computer operation may be preceded by an INITIAL RESET period used to establish initial trial parameters for recursive operations (Secs. 2-4 and 8-14).

The sampling technique of Fig. 7-6*c* is especially useful for reading fast periodic analog-computer solutions into slow recorders, printers, or

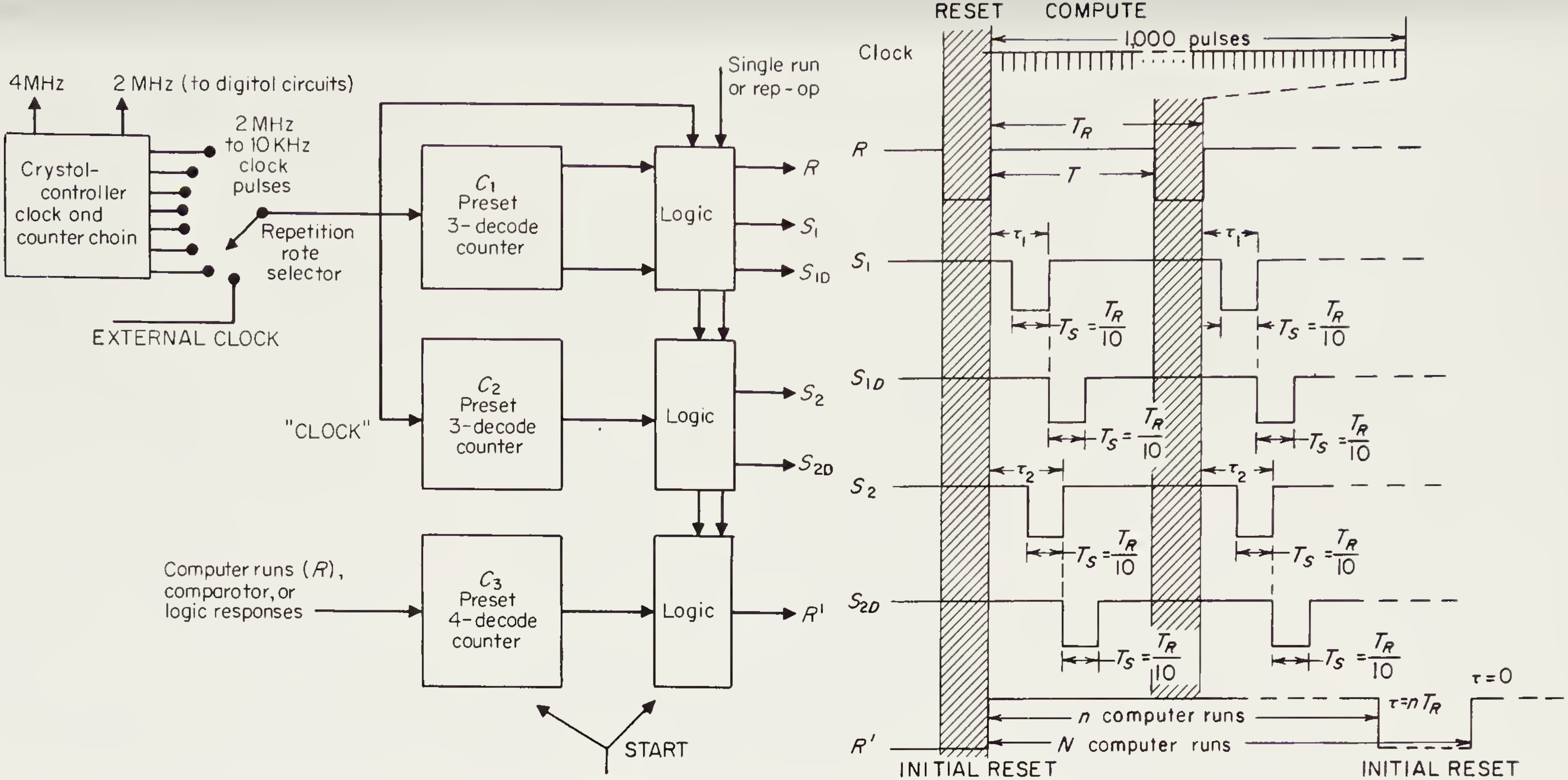


Fig. 7-6*a*, *b*. ASTRAC/LOCUST system for timing computer runs and solution sampling in analog or hybrid computers. A crystal-controlled oscillator and counter chain provide a variety of clock frequencies, which are also available in the digital patchbay. Counter C_1 counts 1,000 clock pulses to define one complete RESET/COMPUTE period (logic outputs R, $\bar{R}$); this can be obtained once by a digital or pushbutton command (SINGLE RUN), or it can be repeated periodically (REP-OP). The analog-computer run time or repetition rate is then determined by the frequency of the clock pulses ("clock") selected by the *repetition-rate switch*, which may also select the integrator capacitors suitable for each time scale. Counter C_1 also provides pulses (not shown) at 10 and 100 times the computer repetition rate, which are often useful. The *clock-pulse counters* C_1 and C_2 can be preset by front-panel switches or digital-computer instructions to produce accurately timed COMPUTE periods T and *sampling pulses* S_1, S_2, S_{1D}, S_{2D}. The *event counter* C_3 counts a preset number of computer runs, comparator responses, or other events for iterative-differential-analyzer programming. (*University of Arizona*, *Refs*. 6, 7, 10, 13, *and* 14.)

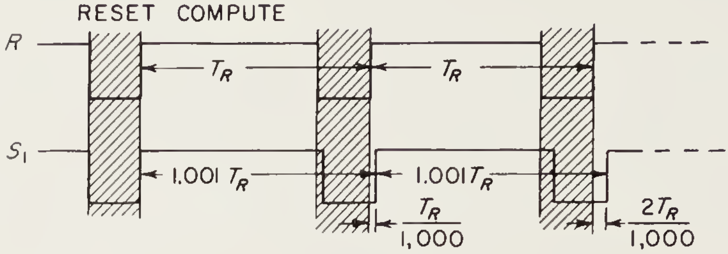


Fig. 7-6*c*. By increasing the delay τ_1 of a readout sampling pulse S_1 during successive COMPUTE periods, we can read sampled-data time histories of periodically repeated analog-computer solutions into slow recorders, printers, or digital computers.

digital computers. Multitrace oscilloscope displays used with repetitive analog computers also can employ digital timing signals derived from the counters to switch oscilloscope multiplexers similar to Fig. 5-15.

(c) Patched Digital Logic (see also Appendix C). Patched gates, flip-flops, and monostable multivibrators connected to analog comparators and electronic switches have been used in many clever iterative-analog-computer setups (Secs. 8-13 and 8-14, Ref. 76). When a digital computer is available, it is convenient to implement as many of the logic and sequencing operations as possible through the digital program. But some patchable logic (say about 40 gates, 30 flip-flops, and a few decimal counters and shift registers for a 150-amplifier installation) still helps to supplement iteration and interface logic and, above all, to breadboard simulated instruments involving digital logic. *Transistor-transistor logic* (TTL) is most readily available, although *emitter-coupled logic* (ECL) introduces much less noise in fast analog circuits (Sec. 7-14). Clock-gated flip-flops present fewer programming hazards, but the most flexible patched-logic arrangement provides flip-flops capable of *both* clocked and nonclocked operation.

Some computers have *monostable multivibrators*, which are triggered to provide a single pulse whose length is controlled by a variable resistor and/or a switchable capacitor. Monostables are simple but not very accurate, and the capacitors must be switched with analog-computer time-scale changes. The writers prefer to mark time intervals with counter circuits slaved to the analog-computer clock, which determines the time scale (Fig. 7-6).

7-4. Patching and Other Programming Systems. (a) Analog-computer Patchbays. General-purpose analog computers are set up through plug-in interconnections with *patchcords.* Direct patchcord connections between jacks on computing-element panels permit a minimum of wiring and shunt capacitance; incomparably better computer utilization, however, is obtained when setups are patched on *removable* *problem boards* (*patchboards, prepatch panels*), which can be plugged into a central *patchbay*. This system permits patching away from the computer, which is then free for other work; and patched programs can be stored for later reuse. Most larger machines have *contact-spring patchbay systems* like those shown in Fig. 7-6.

D-c leakage and crosstalk considerations (Fig. 7-7) make it mandatory to employ *shielded patching systems* with coaxial patchcords in all but the most inexpensive computers, in spite of the added shunt capacitances. Grounded printed-silver grids on plastic patchboards will, however, at least eliminate surface leakage. Shielded systems use all-metal or conducting-plastic patchboards; the use of honeycomb shield structures in plastic patchboards has been largely abandoned as too expensive, although it prevents short circuits from "hot" patchcords touching the shield by accident.

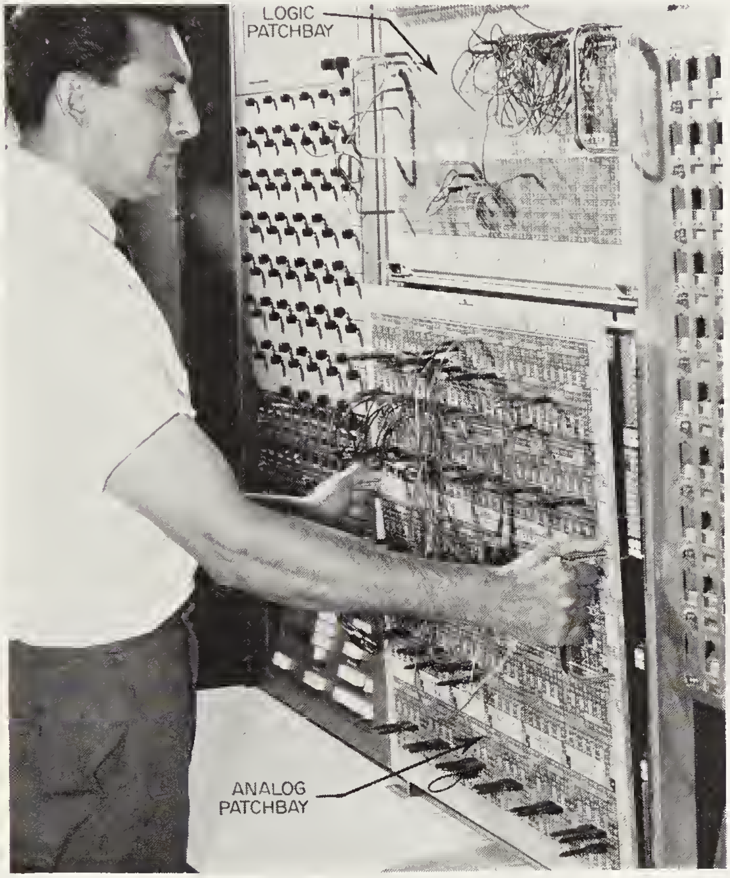


Fig. 7-7. A solid-metal or conductive-plastic patchboard grounds the shields of coaxial patchcords. The patchbay locking lever, or an electric motor, presses the protruding patchcord tips against corresponding contact springs with a self-cleaning wiping action; carefully designed, precious-metal-plated contacts minimize contact resistances. To avoid accidental shorting of "hot" patchcord tips against the grounded patchboard, the operator must deenergize the patchbay before patching. (*Electronic Associates, Inc.*)

To make an inexpensive unshielded patchbay into a shielded patching system suitable for fast analog/hybrid computation:[14]

1. Connect a pattern of unused patchbay springs to ground or power-supply terminations (a-c ground) to serve as shielding between analog terminations
2. Replace the original plastic patchboards with metal patchboards drilled only for the desired analog terminations

Only low-leakage plastic (diallylphthallate) should be employed in analog patchbays.

Patchboards can have pressure-sensitive overlays printed in bright colors to identify terminations. This makes it possible to change terminations, add new components, etc. Large patchboards have to be quite rigid; they must not deform appreciably under the substantial forces required for positive contact of 1,000 to 3,000 patchcord tips.[1]

Good system design minimizes wiring between computing elements and patchbay, for signal wiring and connectors cause capacitive loading, noise, and crosstalk, as well as added initial and maintenance expenses. Computing networks, mode-control relays, electronic switches, diode networks, and solid-state amplifiers are best constructed as separately shielded modules plugged directly into the rear of the patchbay, or module boxes bearing patchbay springs may be bolted into a patchbay receiver. *Patchbay ovens* complicate access to components and are really needed only for high-precision *capacitors.* Note that such capacitors may require recalibration each time the oven is opened (Sec. 3-8) and are therefore best off in an oven by themselves. Ovens can employ blanket-type heating elements to reduce temperature gradients without recirculating fans.

Clever patchbay layout, with a view to short patching connections, improves operating convenience and reduces errors due to patchcord clutter. In larger computations, though, patchcord clutter is practically unavoidable, and one can no longer see the block-diagram structure on the patchbay.

Shielded patching systems are not cheap but cost just under $2 per patchbay termination and per patchcord. On the other hand, high-quality d-c amplifiers are becoming less and less expensive. It follows that *all inverting and output amplifiers associated with multipliers, function generators, resolvers, and digital-to-analog converters should be permanently wired* (*not patched*) *to these computing elements.* This not only saves patching connections but, more importantly, improves bandwidth through short connections and special-purpose equalization.

Hybrid-computer *logic terminations*, which include integrator and track-hold control inputs, multiply/divide and resolver mode control, timing circuits, interface logic, and free logic (Sec. 7-3), do not require shielded patching systems and are usually brought out to an inexpensive unshielded "digital" patchbay.

(b) Setting Coefficients and Functions. Coefficient potentiometers and diode function generators can be set with a digital voltmeter, which can also read amplifier output voltages in RESET and HOLD. For manual potentiometer setting, simple calibration pushbuttons immediately adjacent to each potentiometer (Fig. 3-7) are preferable to relay-selector switching, which requires the operator to check the potentiometer number against the selector keyboard.

Larger analog computers may have *servo-set potentiometers and function generators* controlled individually from a central control panel or from digital storage to permit rapid problem changes. Potentiometers can have individual servomotors, or a motor serves 10 to 40 solenoid-engaged rubber-idler friction drives associated with each potentiometer. Relays operated by a *selector keyboard* place the computer into POTSET, apply a reference voltage to a potentiometer, and connect it into the servo system. The potentiometer is set when its output equals that of an accurate reference DAC controlled by a *coefficient-setting keyboard*, by a *punched-tape reader*, or by a *digital computer*. The potentiometer setting can be checked visually and independently by the computer digital voltmeter. In addition to the coefficient-setting keyboard, the control panel may have a comparison potentiometer permitting *continuous* adjustment of servo-set potentiometers.

Ten-turn potentiometers can be servo-set within 0.02 percent of full scale (20 mV for ±100-volt computers, since the LO terminal is grounded). A typical servo-setting operation takes 1 to 3 sec. A punched-tape system or a digital computer can servo-set 100 potentiometers in less than 6 min.

Digital coefficient units (*DCUs*, *digital attenuators*) are accurate (14- to 16-bit) but relatively slow-setting MDACs (see also Sec. 5-9) for multiplying an analog voltage by a digitally set nonnegative coefficient, which need not be less than unity. DCUs are more expensive ($100 to $250) than servo-set potentiometers ($20 to $50) but *set much more quickly*, *are more reliable*, *last longer*, *and can have better bandwidth*, especially if they are plugged directly into the analog patchbay. Reed-relay-switched DCUs have given way to FET switching (Sec. 5-7). Some DCUs do not employ calibrated resistor networks but are set with a feedback circuit controlled by a precision reference, just like servo-set potentiometers.

Function-generator setting will be further discussed in Sec. 9-8.

(c) Automatic Patching and Time-shared Analog Computation. *Automatic patching systems* replace patchbays and patchcords with relay or electronic-switch interconnections, which can be programmed by a digital computer. The enormous number (5,000 to 50,000, depending on installation size) of on-off switches required amounts, at \$2 to \$5 per switch, to perhaps 25 percent of the total installation cost. Automatic patching may, nevertheless, be attractive, because:

1. Complete repatching requires only 10 to 100 msec, so that *the analog/hybrid computer can be time-shared among several users operating digital-computer-type terminals.* Since ordinary hybrid computers are often idle between computer runs, substantial savings may be realized.

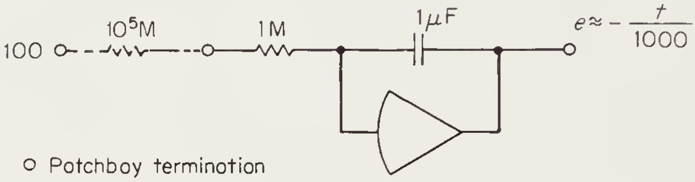


Fig. 7-8*a*. During a typical 100-sec computer run, patchbay leakage and/or patchcord leakage resistances as high as 10^5 *megohms* can cause integrator errors exceeding 0.1 percent of half-scale.

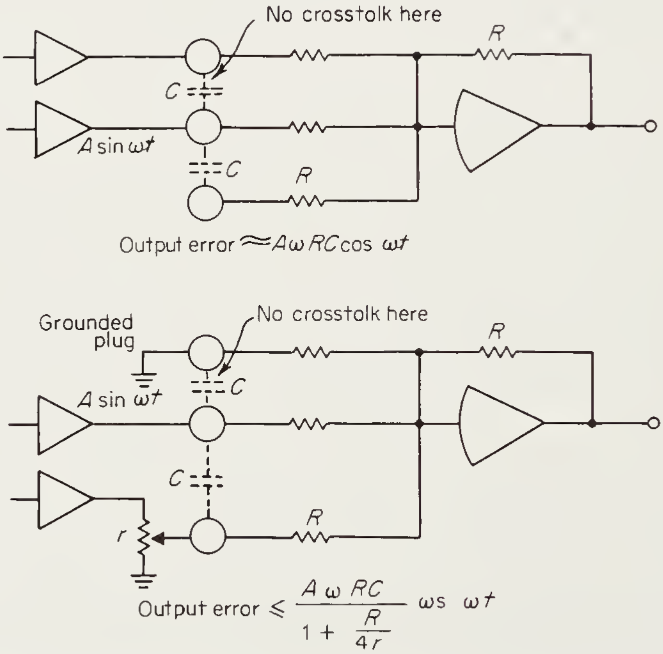

No crosstalk here
R
C
A sin ωt
C
R
Output error ≈ AωRC cos ωt
Grounded plug
No crosstalk here
R
C
A sin ωt
C
R
r
Output error $\leqslant \frac{A\omega RC}{1+\frac{R}{4r}}$ ωs ωt

Fig. 7-8*b*. Patchbay and patchcord capacitances between inputs cause crosstalk into patchbay terminations which are not connected to low-impedance amplifier outputs or ground. Open termination shown contributes $ARC\,\omega \cos \omega t$ volts errors (0.3 percent of half-scale for $C = 5$ pF, $R = 100$K, $F = \omega/2\pi = 1$ KHz). Grounding unused input terminations reduces crosstalk but also amplifier bandwidth. Termination capacitances to ground (not shown) will reduce the crosstalk, typically to about one-half of the values given in the figure.

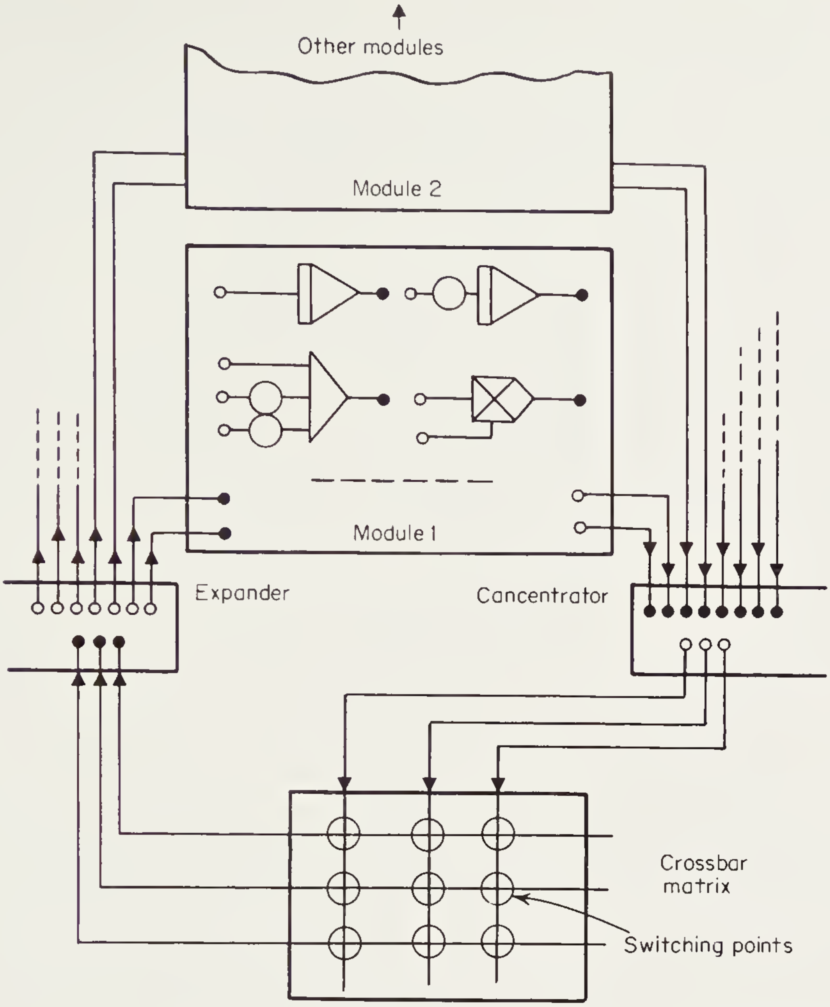


Fig. 7-9*a*. Automatic-patching-system design (simplified). In each functional group or module (only one is shown in some detail), each white input terminal can be connected to any one black output terminal. A one-level crossbar matrix is used; each matrix output can connect to a single input, while matrix inputs can connect to any number of outputs.

2. Patchboards and patchcords for problem storage are not cheap (about $900 for a typical problem), so that punched-tape storage of 10 to 20 problems may pay for automatic patching.
3. Fast problem changing saves engineering time and overhead.

Optimal use of automatic patching systems requires *special digital software*, viz., an "analog compiler" such as APACHE,[18,19] APSE,[29] or ACTRAN[30] to generate the interconnection pattern from a set of differential equations, plus a diagnostic program for checking the necessarily large number of relays or electronic switches automatically.

It is not necessary to provide "exhaustive" switching of every computing-element output to every input. Instead, one divides the available computing elements into (largely similar) *functional groups* or *modules* chosen so that typical problems place most connections *within* modules.

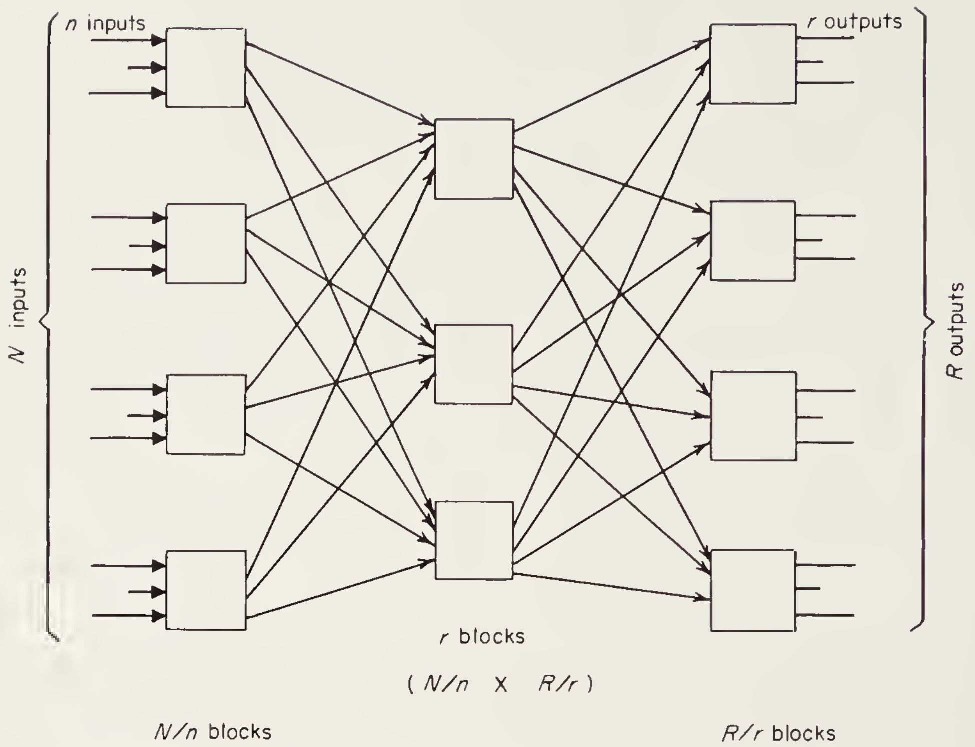


Fig. 7-9*b*. A three-level switching-matrix system. (*Based on Ref.* 25.)

A fairly comprehensive set of possible interconnections is then provided within each module, while *intermodule* connections are made through a limited number of *trunks*. A trunk line can connect to a number of module outputs through a selector switch (*concentrator*) and/or to a number of module inputs through an *expander* switch. To reduce the number of trunks needed, trunks can be connected through *crossbar* (*matrix*) *switches* (Fig. 7-9*a*), possibly arranged in several levels (Fig. 7-9*b*). In addition to these interconnections, some extra switches connect limiter diodes and integrator capacitors, switch multiplier and resolver modes, etc. (see also Refs. 79 and 80).

References 23 to 26 present formulas for the number of switches required with different dimensions of functional groups, concentrators, expanders, and crossbar matrices. The designer attempts to minimize this number subject to constraints on the size and design of the functional groups, which depend on the class of problems usually solved; the number of probable intermodule connections is another constraint. Intermodule connections may or may not be restricted to adjacent modules.

As an example, a functional group in the Applied Dynamics AD-4 computer (Fig. 7-2) comprises 16 summer/integrators and 12 multipliers and is digitally programmed through only 2,184 FET switches plus 64 digital coefficient units. Economy in the use of switches was realized by clever restriction of possible interconnections. Connections between each pair of functional groups require another 768 FET switches. Hence a system comprising three modules (48 summer/integrators, 36 multipliers) would require $3 \times 2{,}184 + 3 \times 768 = 8{,}856$ switches.

Reference 28 describes a solid-state crossbar matrix based on the multiplexer circuit of Fig. 5-15*b*.

To limit the number of switches needed, automatic patching can very well be applied to only part of a computer, while setup portions requiring less frequent changes are manually patched. Applied Dynamics "time-sharing" terminals, intended mainly for educational purposes, are an example; each time-sharing terminal switches only a few connections, sets a few coefficients, and includes a storage-CRT display.

7-5. Readout Systems. (a) Analog-computer Readout. The most accurate way to extract numerical data from analog-computer solutions is by sampled-data digital readout (Sec. 7-3*b*). In practice, however, we require insight into an entire solution rather than discrete numerical data; and immediately available, storable graphical computer-solution records are an indispensable machine-man link. This is true even though the accuracy, speed, cost, and maintainability of electromechanical recorders leave much to be desired. Direct-writing *strip-chart recorders* (Fig. 7-10) capable of recording two, four, six, or eight variables against time are still the principal means for recording computer solutions; simultaneous presentation of multiple records can provide very valuable insight into the operation of a simulated system. Strip-chart recorders are inaccurate, slow, and expensive. *Servo plotting boards*, while more accurate, are even slower and still more expensive. Light-beam-galvanometer recorders, with or without direct-writing features, are faster (useful up to several kilohertz), but are also inaccurate and expensive. Multistylus and facsimile recorders are too slow.

Even in "slow" analog computers, *oscilloscope displays* are useful for preliminary checking and scaling runs as well as for servicing; they also produce permanent Land-camera records. Oscilloscope voltage and time scales should duplicate those of the recorders used with the computer.

(b) Strip-chart Recorders. Strip-chart-recorder penmotors are either galvanometer movements or miniature a-c servo actuators. Permanent records are produced either with ink fed through tubular recorder pens by gravity or pressure (aerosol can), or on heat-sensitive paper contacted by heated ridges on each pen along a sharp edge perpendicular to the paper feed (Fig. 7-10*b*). The latter method inherently converts the angular pen motion into a rectilinear displacement along the contact edge; in ink recorders, a special pen linkage is required to produce rectilinear-displacement records. Although not more accurate, records with rectilinear coordinate lines are much more pleasing.

A recorder penmotor is, essentially, a second-order dynamical system with mechanical inertia, spring stiffness supplied either by a spring or by the penmotor-servo stiffness, and damping through friction together with equalization in the penmotor drive or amplifier circuit. Damping is usually adjusted for a slight overshoot. The best strip-chart recorders claim accuracies between 0.5 and 1 percent of half-scale up to a few hertz, with useful output at reduced accuracy up to between 100 and 150 Hz. Ink recordings are crisper, use more durable paper (heat-sensitive paper is pressure-sensitive as well), and their trace width does not change with pen-displacement rate and paper speed; they also tend to be somewhat faster than heat-writing recorders. Heated-pen recorders, however, do away altogether with the recurring nuisance of filling and maintaining ink-supply systems.

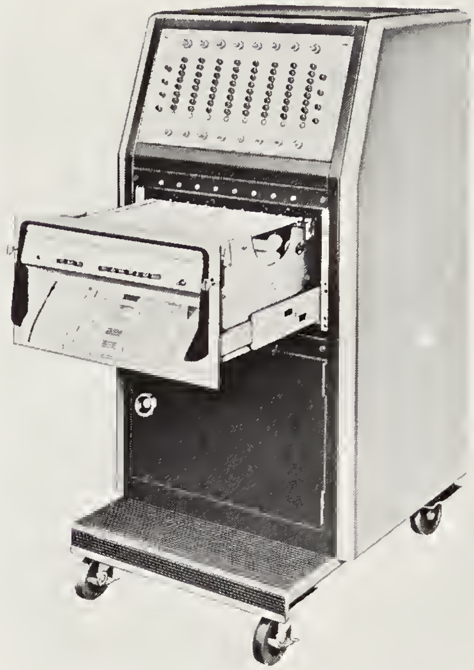


Fig. 7-10*a*. Multiple-channel strip-chart recorders can plot several variables against time and are the most generally used analog-computer readout devices. (*Brush Instrument Division, Clevite Corp.*)

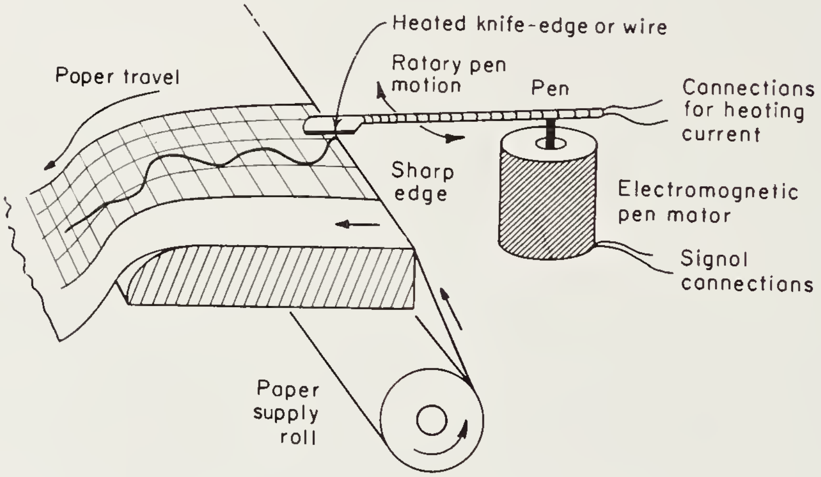

Heated knife-edge or wire
Paper travel
Rotary pen motion
Pen
Connections for heating current
Sharp edge
Electromagnetic pen motor
Signal connections
Paper supply roll

Fig. 7-10*b*. Heated-pen strip-chart recorder. The heated edge of each pen contacts the heat-sensitive recorder paper along a sharp edge perpendicular to the direction of paper travel and produces rectilinear-coordinate records.

Gear-shift levers or remotely controllable solenoid systems permit a range of paper speeds between 0.05 and 200 mm/sec. The paper feed is driven by a line-frequency synchronous motor or, preferably, by a tuning-fork-oscillator-controlled motor. A chart-speed accuracy within 0.25 percent is reasonable. In many analog-computer installations, recorders can be controlled from the main console, which may provide for starting and stopping, paper-speed changes, and remotely controlled calibration inputs

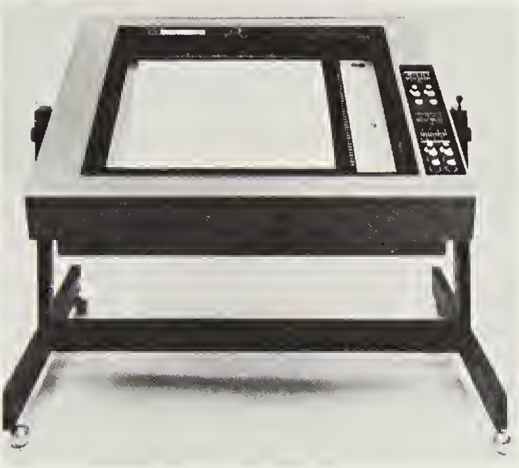


Fig. 7-11*a*. A large servo plotting board permitting vertical, horizontal, or tilted operation. (*Computer Systems, Inc.*)

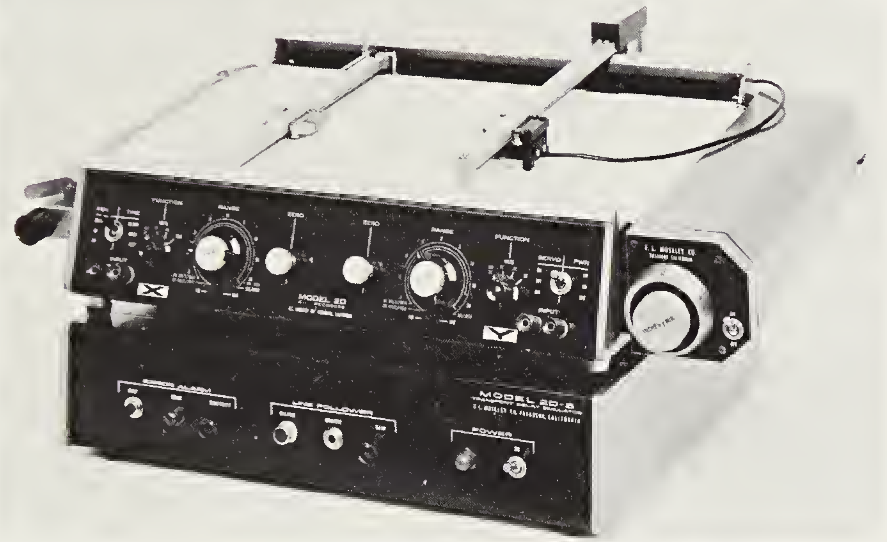


Fig. 7-11*b*. Tabletop *xy* recorder with chart-drive and photoelectric curve-follower accessories set up as a transport-delay simulator. In this application, the x servo is not used, but a chart drive feeds paper at 2, 4, 8, 16, 24, or 32 in./sec. The servo-positioned pen records $f(t)$, and the curve follower reads $f(t - \tau_D)$, with delays τ_D between 4 sec and 7 min determined by the adjustable distance between pen carriage and curve follower and by the paper speed. With the pen replaced by a second curve-follower head, the device also reads $f(t)$ and $f(t - \tau_D)$ from a given curve for correlation measurements. With the time-delay accessory removed, the recorder functions as a normal *xy* recorder or curve follower. (*Moseley Division, Hewlett-Packard Corp., Pasadena, Calif.*)

(zero, ±10 volts, and/or ±100 volts). It is also possible to print computer-run numbers on the recorder track. It is very convenient to have all recorder voltage and speed scales duplicated on a checkout oscilloscope, since preliminary oscilloscope runs to check scales and presentation can save a great deal of chart paper.

Modern computer d-c amplifiers can drive recorder penmotors directly, but most recorders contain their own d-c amplifiers complete with output-current limiters and equalizing networks producing a desired pen-displacement step response.

(c) Servo Plotting Boards and Curve Followers. *Servo plotting boards* (*xy recorders*) produce inked plots of a computer variable $y(t)$ against another variable $x(t)$ or against the time with the aid of a servo-positioned pen (Fig. 7-11). The larger plotting boards may have two pens with separate y-servo drives on the same x carriage to record two functions $y_1(x)$ and $y_2(x)$; a relay comparator interchanges the roles of the pens whenever the two curves intersect. Servo plotting boards can have static accuracies within 0.1 percent of half-scale, but they are quite slow, with typical output rates and accelerations respectively below 20 in./sec and 750 in./sec^2, and 5-deg phase shift for 3-Hz 8-in.-peak-to-peak sine waves. In some recorders, the y displacement (pen) can respond more quickly than the x displacement (pen carriage). As with strip-chart recorders, preliminary oscilloscope runs can save scaling troubles.

Most xy recorders can be fitted with photoelectric* *curve-follower heads* which are connected to the y servo to follow an inked curve $y = f(x)$ as the pen carriage is driven in the x direction, so that the y-follow-up potentiometer produces an output voltage $y = f(x)$. Curve followers will not be misled by suitably colored coordinate lines and are a very convenient means for entering low-frequency data into computations.

Most servo tables can be obtained with internal time-base circuits or time motors to record y against the time with a variety of time scales. Servo tables fitted with multiple pen carriages and a paper-feed drive can record $f(t)$ and read $f(t - \tau_D)$ to produce fixed or variable *transport delays* (Fig. 7-11*b*; see also Sec. 2-19).

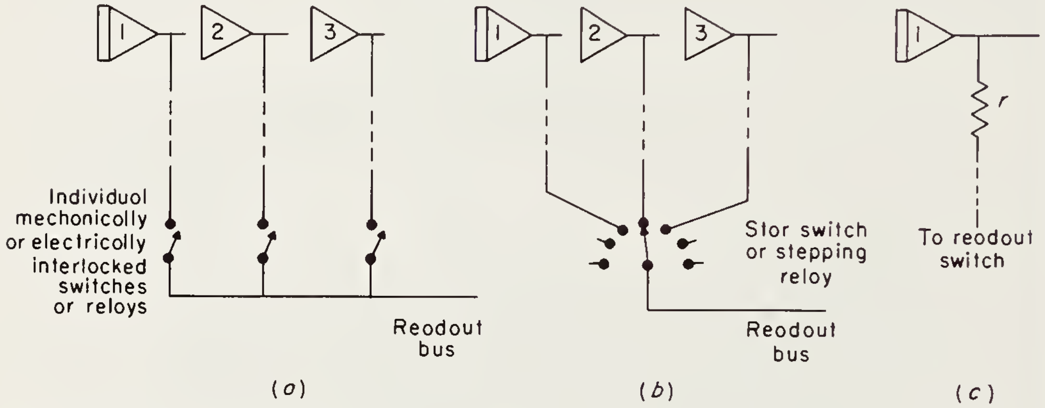

1
2
3
Individuol mechonicolly or electricolly interlocked switches or reloys
Reodout bus
(o)
1
2
3
Stor switch or stepping reloy
Reodout bus
(b)
1
r
To reodout switch
(c)

Fig. 7-12. Readout scanning circuits. Circuit (*a*) employs separate switches or relays and loads amplifier 2 with nC_{SWITCH} plus wiring capacitance. Circuit (*b*) produces the smaller load C_{SWITCH} plus wiring capacitance. Resistance r (10 ohms to 1K) in (*c*) "stops" amplifier phase shift due to readout-circuit capacitances at the expense of a simple calibration change in d-c readout instruments; the high-frequency response of oscilloscope and sampling readout circuits will be impaired, although equalization-network correction may be possible.

(d) Amplifier-readout Switching. Recorder connections are usually patched, but a system of switches or relays connecting a selected amplifier output to a readout voltmeter and/or oscilloscope is convenient for quick checks of computer setups, solutions, and amplifier balance. Stepping relay, reed-relay matrix, or crossbar switching systems require maintenance and are justified only with automatic or semiautomatic checking and setup systems (Sec. 7-6). The simplest readout selectors are inexpensive rotary switches. They are inconvenient to use with more than a few amplifiers; interlocking pushbuttons are more convenient, but a system of n parallel-connected switches loads the readout circuit with $n - 1$ switch capacitances (Fig. 7-12*a*, *b*). *We can, however, practically eliminate readout-circuit capacitance effects on amplifier bandwidth and stability by inserting an accurately known* 100-*ohm to 5K resistance between each amplifier output and its readout connection*

---

* Earlier inductive, capacitive, and resistive curve followers could only follow special conducting-ink traces and may be considered obsolete.

(Fig. 7-12*c*). Readout voltmeters are readily recalibrated to allow for the added resistances In "slow" analog computers, even oscilloscopes and recorders can be suitably recalibrated and reequalized.

7-6. Calibration and Program Checks.

(a) Calibration and Amplifier-balance Checks. Since most chopper-stabilized d-c amplifiers will overload briefly when high voltage is switched off and on, the POTSET/BALANCE CHECK mode (see also Sec. 7-3*a*) of most analog computers removes only the computer reference voltages from patchbay and diode function generators to permit safe patching; in this condition, outputs of all properly functioning computing elements read zero. A *balance-check relay or pushbutton* in each d-c amplifier can decrease the feedback ratio, so that amplifier offset above a few millivolts will light the overload light without causing too serious an overload. The amplifier is then balanced so as to extinguish the overload light, by reference to a balance meter, or for minimum chopper noise.

A conveniently located set of *calibration pushbuttons* operative in POTSET or RESET makes it easy to check the positive and negative reference voltages and the recorder calibration and null against the digital voltmeter.

(b) Overload-alarm Systems and Minimum-excursion Indicators. *Overload-alarm systems* inform the operator that an amplifier overload due to scaling or malfunction has made the current computer solution meaningless. The simplest *overload indicators* for ±100-volt computers are simply 60- to 80-volt neon lights connected across each amplifier output through voltage-divider networks. *Error lights* at the outputs of d-c amplifier chopper-stabilizer channels (Sec. 3-36) indicate excessive d-c amplifier summing-point voltages and therefore detect not only voltage overloads but also current overloads and a variety of amplifier malfunctions. Error lights are usually preferred in "slow" analog computers. Error lights will, however, miss overload peaks too short for the low-pass chopper-stabilizer channels and are therefore not appropriate for overload detection in repetitive analog computers, where our main interest is in transient voltage overloads.

Since the operator might overlook a momentary overload in one of a large number of amplifiers, the individual overload indicators are often connected to a common *master overload-alarm circuit*. The latter operates a light, an audible alarm, and/or digital logic which places the analog computer into HOLD or RESET, sets an overload flip-flop (*stored overload*), or interrupts a digital computer.

A *minimum-excursion indicator* (MEI)[3] indicates that a corresponding amplifier output voltage has exceeded, say, 10 percent of its total dynamic range in the course of a computer run. MEIs show up excessively low scale factors; with proper scaling, all MEI lights must be extinguished at the end of each computer run.

Figure 7-13 shows a possible design for a ±10-volt overload indicator or MEI.[3]

(c) Static Checks, Rate Tests, and Time-scale Check.[3] In the RESET mode, the output voltages of potentiometers, summers, multipliers, etc., must be specified functions of the integrator initial-condition settings. *Static checks* of analog-computer programs check these output voltages to detect programming errors or malfunctions. Improved static-checking systems also permit one to switch each integrator summing junction into a summing-amplifier circuit whose output is read out to indicate the *initial rate of change* of the integrated variable. Furthermore, since ordinary static checks cannot properly check points whose initial voltage is zero, one may introduce *artificial initial conditions* during a preliminary static check with the true initial conditions. Similar tests can also be applied in mid-problem with the computer in the HOLD condition.

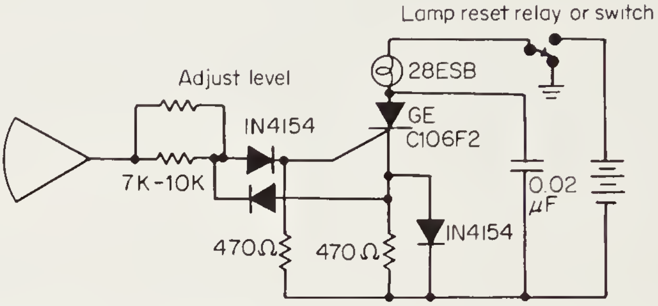


Fig. 7-13. A ±10-volt overload-indicator circuit with visual and logic outputs, using a silicon controlled rectifier. (*University of Arizona, Ref.* 14.)

In an alternative static-checking scheme (PROBLEM CHECK STATIC mode in Reeves Instrument Corp. computers), a relay disconnects each integrator output from the computer setup and substitutes a test voltage. The same relay shunts a feedback resistor across each integrating capacitor, so that the integrator output can be used to read the input sum.

To check integrator and mode-control operation, a variety of *rate-test* circuits permit each integrator to integrate a known input for a known time interval. The computer is then switched into HOLD, and integrator outputs are read.

A rapid check for effects of insufficient computing bandwidth is to repeat the solution at one-half the original time scale. Large analog computers may provide a *time-scale-check* switch which doubles all integrating capacitors and halves the recorder speed through appropriate relays.

With modern hybrid computers, suitable digital-computer programs precompute static-test results, perform the complete static check automatically, and produce accurate records of the test results (see also Sec. 9-11).

7-7. Analog-computer Accuracy Checks and Maintenance Procedures. Effective computer utilization requires a preventive-maintenance program to minimize operating failures and computer down time; it is also desirable to check component accuracies periodically. Depending on the size and workload of the computer installation, one performs daily, weekly, and/or monthly checks on:

Power-supply and reference voltages
Resistor and potentiometer continuity
Calibration of readout devices
Amplifier gain, noise, and offset
Amplifier input gains
Integrating-capacitor calibration (rate test, Sec. 7-6*c*)
Operation of servo and electronic multipliers (triangle test, Sec. 6-15)
Operation of function generators, diodes, and relays
Digital-module operation
Spikes and offset in electronic switches
Recorder ink and paper supplies

Suitable test circuits are wired permanently on special test patchboards substituted for the computer problem boards; Refs. 32 to 43 list many such circuits (see also Chaps. 3 and 6). Defective or inaccurate components are unplugged and replaced. It is desirable to have about one spare amplifier in 20.

Hybrid computers can, again, automate maintenance test procedures with suitable digital-computer programs for testing analog, logic, and linkage circuits (see also Sec. 9-16).

Proper maintenance procedure must include a filing system for *failure reports* and *service reports* designed to locate sources of frequent failures for eventual circuit modification, to ensure a proper supply of spare parts, and to establish responsibilities.

DIGITAL COMPUTER AND INTERFACE

7-8. Survey: Digital-computer and Linkage Requirements.[49-51] The data and control-logic interchange between a digital computer and an associated hybrid-computer linkage is rather similar to that for a digital computer supervising an industrial process, so that many digital computers designed for process-control applications and real-time supervision of experiments, etc., are suitable for hybrid computation. The digital computer will require a *parallel input/output bus* (I/O bus) for programmed operations (Secs. 7-9 and 7-10), a *priority-interrupt system* with between 4 and 128 interrupt lines (Sec. 7-11), and, if possible, also a *direct-memory-access channel* for fast block transfers of data (Sec. 7-12).

16- and 18-bit minicomputers with 8 to 16K memories are perfectly suitable for "housekeeping" operations such as coefficient setting, static checks, and analog diagnostics; such small computers can also sequence analog operations, process and print results, and serve for simple iterative optimization and statistics, plus some function-generation and time-delay simulation. They can even do some combined simulation if an accessory floating-point-arithmetic unit is added or if a block-oriented, fixed-point simulation language is available (Sec. 9-15). For serious combined simulation, however, much more expensive medium-size machines with floating-point arithmetic and moderate-size memories (16K to 32K) are most common. Very large scientific and general-purpose digital computers are time-shared and even dedicated for hybrid computation in large installations, but such machines are not always designed for real-time work and usually require an auxiliary smaller digital computer (peripheral processor) for efficient time-shared operation.

Interface operation and programming for typical applications will be described in Secs. 9-11 to 9-17. We now list *the hardware functions of a general-purpose hybrid-computer interface* linking a typical 150- to 250-amplifier analog computer to a medium-size digital machine (Fig. 7-14).

1. 4 to 40 channels of *analog-to-digital conversion*. Except with very slow analog variables (say below 10 Hz), a multiplexed ADC will require a sample-hold circuit for each channel. Most installations multiplex a fast (5 to 20 μsec) 13- to 14-bit ADC, but this should be weighed against the possibility of using separate slower (and less expensive) ADCs, which save the cost (and errors) of multiplexer and sample-hold circuits, provide separate buffers, and simplify the interface design. In combined simulation, 20 separate 200-μsec ADCs are just as fast as one multiplexed 10-μsec ADC.[14,50]
2. 4 *to* 120 *MDACs*. Only double-buffered MDACs are recommended, unless hold circuits of the type shown in Fig. 9-3 are provided; in some applications, fast MDACs can be replaced by more digital coefficient units (DCUs).

ADCs and MDACs ought to interface with direct-memory-access data channels as well as with the program-controlled I/O bus (Secs. 7-9 and 7-12).

In addition to data conversion, the interface must implement essential *control logic and timing*. Some of the following functions can be hardwired, but it is convenient to have most of them terminated in the hybrid-computer *digital patchbay:*

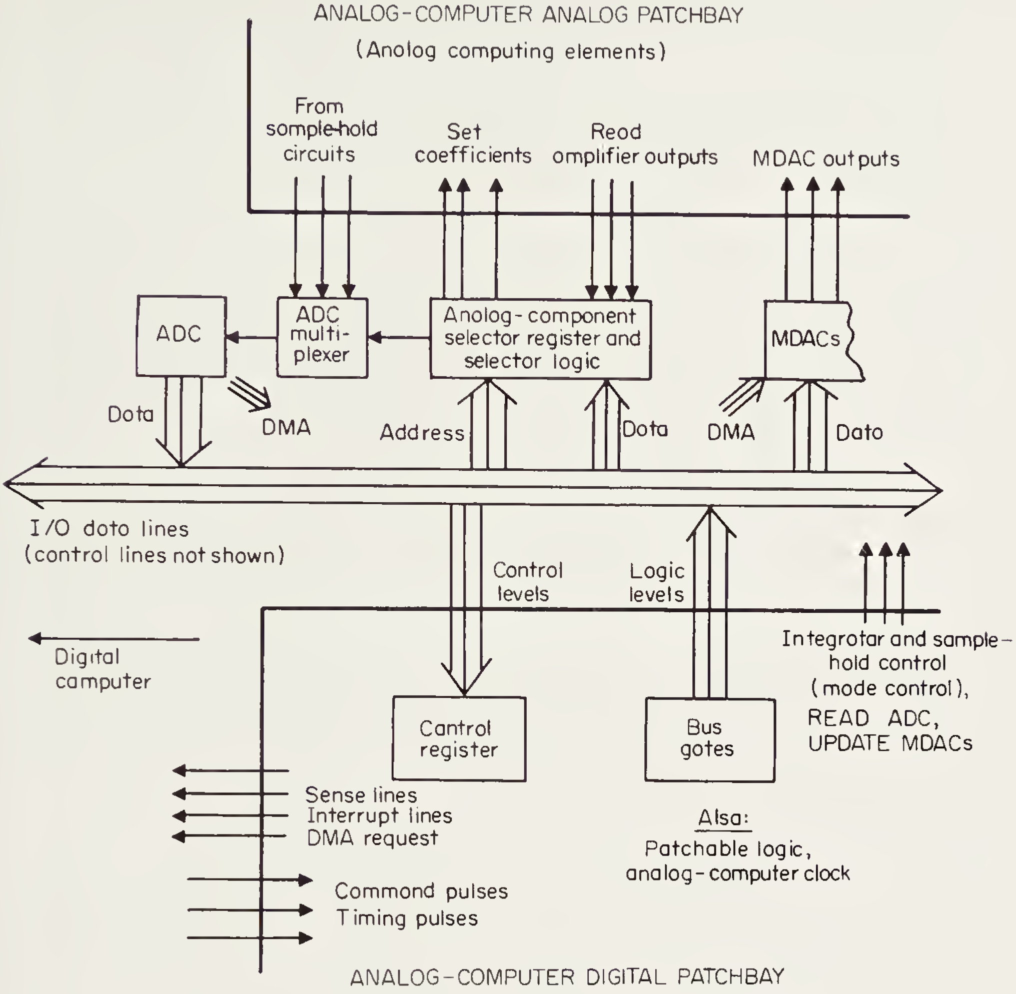


Fig. 7-14. A general-purpose hybrid-computer linkage.

3. A patchable 8- to 24-bit *control register* (or one patchable and one hard-wired control register) loaded by the digital computer via the I/O bus can place the analog computer, or selected integrators, into RESET, COMPUTE, and HOLD; change integrator capacitors; and operate any other analog-computer switches through patched digital logic.
4. A few digital-computer *command pulses* (IOT or IO pulses) and *subdevice-bit levels* corresponding to specific I/O instructions (Sec. 7-9*b*).

Discrete-signal terminations leading from the analog to the digital computer include:

5. 8 to 18 *bus gates* (Sec. 7-9*c*) for reading analog-computer logic levels into the I/O bus.

6. 4 to 20 *sense lines* (Sec. 7-10) which can be interrogated by digital-computer instructions to read analog-computer logic levels ("flags").
7. 4 to 32 *interrupt lines* (Sec. 7-11).
8. Terminations for requesting *direct-memory-access* data transfers (Sec. 7-12).

Some sense and interrupt lines can be hard-wired to signal ADC CONVERSION COMPLETED. The analog-computer *master overload signal* (Sec. 7-6*b*) can be either patched or hard-wired to an interrupt line.

Some interface functions, such as ADC READ and double-buffered-DAC UPDATE commands (Sec. 7-9*c*) are best ORed to both a hard-wired processor command *and* a patchbay termination.

The remaining interface function is the addressing of perhaps up to 1,000 *selector switches or relays* used to set coefficient potentiometers (Fig. 3-7*a*) and to read amplifier outputs into an ADC or digital voltmeter (Fig. 7-12). Logic circuits controlling each switch decode a selection code set into an *analog-component selector register*, which can be loaded from a keyboard as well as through the digital-computer I/O data bus. The selector register may also control the ADC multiplexer, or the latter may have a separate address register. More elaborate linkages also permit the digital computer to *read* the code in the selector register.

We will next discuss the various interface functions in more detail.

7-9. Programmed I/O Operations.[50] **(a) The Party-line I/O Bus.** Most process-control and hybrid computers transmit digital data on

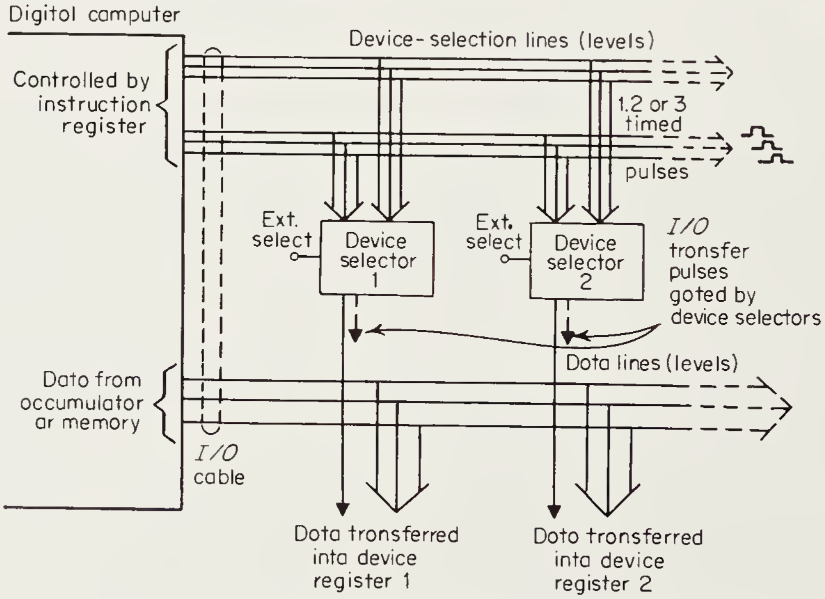

Digital computer
Device-selection lines (levels)
Controlled by instruction register
1,2 or 3 timed pulses
Ext. select
Device selector 1
Ext. select
Device selector 2
I/O transfer pulses gated by device selectors
Data lines (levels)
Data from accumulator or memory
I/O cable
Data transferred into device register 1
Data transferred into device register 2

Fig. 7-15. Programmed control of multiple devices by a digital computer with party-line I/O bus. An I/O instruction addressed to a specific device is recognized by a device selector, which gates data-transfer or command pulses to the device in question. (*Based on Ref.* 78.)

parallel 8- to 18-bit buses; i.e., all data bits are transmitted simultaneously in the interest of processing speed. Serial data transmission is usually restricted to communication links.

The 8 to 18 data-bit lines can be *bidirectional* (this takes extra logic at each device), or we may have *separate input and output buses* (this takes less logic but more interconnections). In addition to the data lines, we will need a comparable number of interface-control-logic lines. Provisions for switching (multiplexing) all these connections to as many as 1,000 or as few as three peripheral devices could compromise our processor design. Thus, most interface systems employ a *party-line I/O bus* of the general type illustrated in Fig. 7-15. Here, all "devices" (printers, displays, ADCs, DACs, etc.) intended to receive or transmit data words are wired to a parallel I/O *data bus* connected to a processor register via suitable logic. Additional party-line wires carry *control-logic signals* for selecting a specific device and its function (e.g., transmission or reception), and also synchronize data transfers with the digital-computer operating cycle.

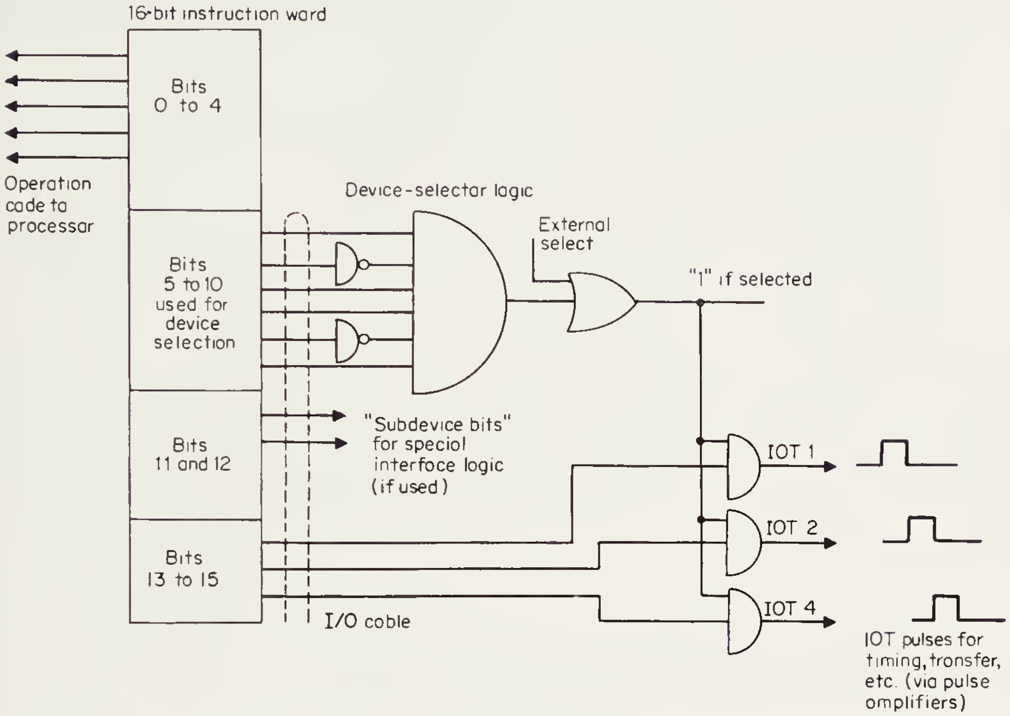

16-bit instruction word
Bits 0 to 4
Operation code to processor
Bits 5 to 10 used for device selection
Device-selector logic
External select
"1" if selected
Bits 11 and 12
"Subdevice bits" for special interface logic (if used)
Bits 13 to 15
I/O cable
IOT 1
IOT 2
IOT 4
IOT pulses for timing, transfer, etc. (via pulse amplifiers)

Fig. 7-16. Program-controlled selection (decoding) of device address and device functions. (*Based on Ref.* 50.)

(b) Program-controlled Device Selection and Operation. For a minimum of linkage hardware, interfaces work with programmed digital-computer instructions (see Sec. 7-12 for direct-memory-access operation). Figure 7-16 shows how the individual bits of an input/

output instruction word in a typical processor determine device-selection and control-line signals on a party-line I/O bus:

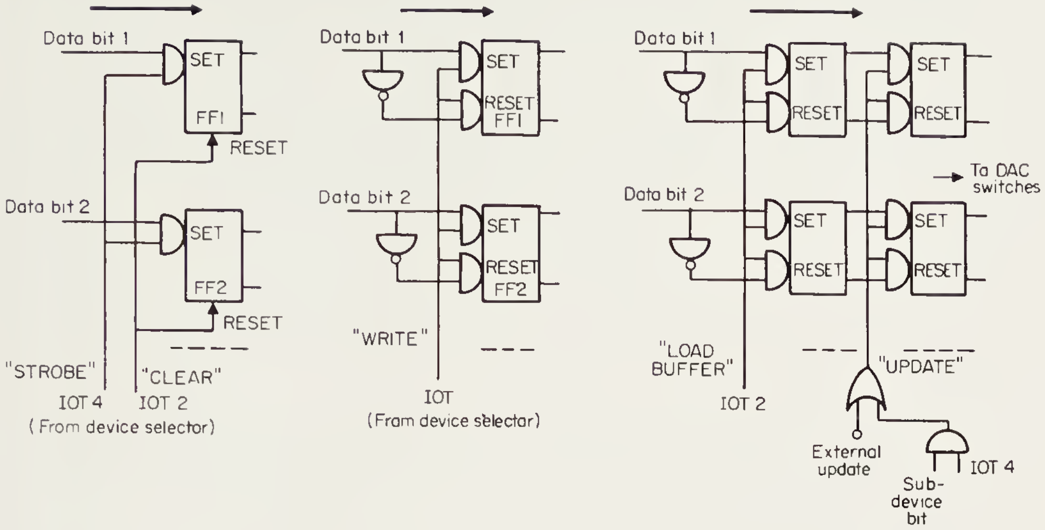
1. Bits 0 to 4 tell the processor that an I/O instruction is wanted. One of these bits can select a READ or WRITE operation, or this decision may be left to a logic input from the device.
2. Bits 5 to 10 place logic levels (0 or 1) on *device-selection lines* parallel-connected to all devices on the I/O bus. When these lines carry the *device-selection code* assigned to a specific device, its *device selector* (decoding AND gate) accepts (and regenerates) a set of one, two, or three *successive command pulses* (IOT pulses) used to effect data transfers and other operations in the selected device, as determined by instruction bits 13 to 15.
3. Bits 11 and 12 ("subdevice bits") determine levels on two more control lines. These can be used to select additional devices or different functions to be performed by a given device.

The arrangement of Fig. 7-16 permits selection among $2^{11} = 2{,}048$ functions corresponding to different combinations of IOT pulses and subdevice bits. This system requires *four complete digital-computer machine cycles* for each I/O operation, one to fetch the instruction, and one for each IOT pulse. Many modifications of our basic scheme are possible:

1. Different numbers of processor-code bits, device-selection bits, subdevice bits, and IOT pulses may be used.
2. IOT pulses can be simultaneous (on different lines), rather than successive, to save execution time.

To save interconnections at the expense of time and hardware, some I/O systems employ their *data lines* to transmit device address and function in a first instruction; data transfers then require a second instruction (note that not all I/O instructions transmit data). Quite in general, devices with more different functions than we can control with subdevice bits and IOT pulses can be designed to accept, store, and execute multibit "device instructions" transmitted over the I/O-bus data lines by I/O instructions. Examples of such superdevices are data channels (Sec. 7-12), displays, and the analog-component selector system of Fig. 7-12.

(c) Data Transfers and Other Device Operations.[50] The most common application of the device-selector-gated command pulses is *data transfer* from and to the processor; note that the pulses are synchronized with the processor operation cycle, and thus with the processor's ability to transmit or accept data. Figure 7-17 illustrates the principal data-transfer techniques.



(*a*) (*b*) (*c*)

Fig. 7-17*a* to *c*. Transfer of numerical data or control-logic bits into a device register: clear and strobe (*a*), jam transfer (*b*), and double-buffered DAC interface (*c*) (see also Appendix B).

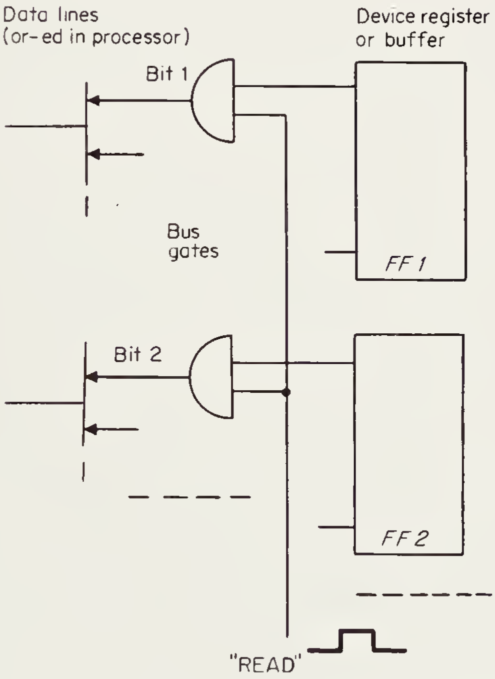


Fig. 7-17*d*. Writing device data or control logic into the digital computer.

1. *Clear-and-strobe Transfer* from I/O bus parallel data lines into the flip-flops of a device register (Fig. 7-17*a*). Each flip-flop is first cleared by IOT2; then IOT4 strobes the 1s on the data bus into the flip-flop register.
2. *Jam Transfer* (Fig. 7-17*b*). A single command pulse (IOT4) sets or resets the device-register bits in accordance with the data-bus levels. Jam transfers require only one pulse period and *must* be used whenever the register resetting operation would disturb device functions. This is true, for instance, with DACs required to have a continuous voltage output and also with *analog-computer logic-control registers*, which must continuously establish computer modes.
3. *Double-buffered-register Transfer* (Fig. 7-17*c*). Data are transferred into the buffer register by either a clear-and-strobe or jam-transfer operation and are then jam-transferred into the device register. Double-buffered DACs permit simultaneous transfer ("updating") of the analog output of all DACs in an analog computer.
4. *Reading from a Device into Processor*. In Fig. 7-17*d*, the correctly timed IOT2 pulse gates data from a device into a processor register such as the accumulator via the I/O data lines.

Device-selector-gated processor command pulses (IOT pulses) are not only used to transfer data. Command pulses can also themselves *set and reset flip-flops* and *increment or decrement counters* in device interfaces.

7-10. Sense-line Operation.[50] The IOT pulses implement program-controlled operations at the correct processor-timed instants. But a device thus addressed might not be ready; a very important example is an ADC which has not completed a conversion. In such cases, program-controlled data or control-logic transfers may be preceded by a *sense-line interrogation or flag test*. The device status is indicated by a *flag* (logic level, usually a flip-flop output). A special instruction addressed to the associated device-selector gates one of the command pulses (IOT1 in Fig. 7-18) into a *sense gate* and, if the flag level (sense line) is up, onto the *skip bus* in the interface cable. The pulse on the skip bus then increments the processor instruction counter, which skips the next instruction to produce a program branch. An example would be:

1. SKIP IF ADC FLAG IS UP
2. REPEAT LAST INSTRUCTION
3. READ ADC

The program will cycle until the device flag is up, at which time the program continues, usually with a data-transfer instruction.

Other sense-line systems exist. Many computers do away with device-selector addressing of sense gates by simply accepting flag levels on multiple sense lines, which are interrogated with processor instructions like SKIP IF SENSE LINE 5 IS UP.

Instead of interrogating multiple sense lines, one may have a special READ FLAGS instruction which gates several flags onto the *data bus*. The resulting "status word" can then be logically interpreted by the computer program; in particular, the data word may serve as an indirect address for the next instruction and thus permit multiple branching.

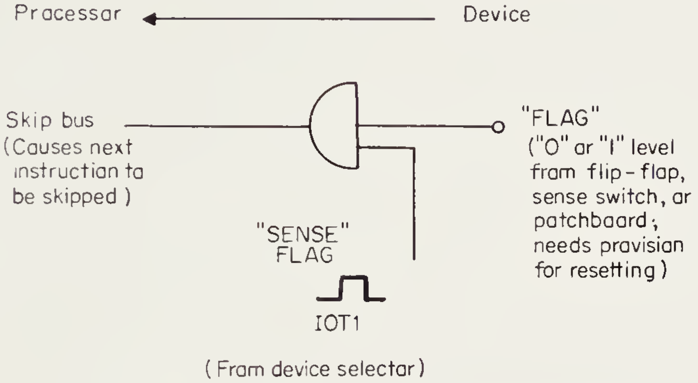

Processor
Device
Skip bus
(Causes next instruction to be skipped)
"FLAG"
("0" or "1" level from flip-flop, sense switch, or patchboard; needs provision for resetting)
"SENSE" FLAG
IOT1
(From device selector)

Fig. 7-18. Sense-line operation using a device selector, sensing gate, and common skip bus.

Sense lines are inexpensive, but even small digital computers can rarely afford the time for "idle" sense/skip loops such as the simple ADC example shown above. Sense lines are therefore used mainly for decisions between device-dependent program branches which do *not* cause idling. Otherwise we require *program interrupts*.

7-11. Interrupt-system Operation.[50] In an *interrupt system*, a device-flag (level) indication interrupts the computer program after the current instruction, causes the program counter and selected processor-register contents to be saved (in memory or spare registers), and starts a new instruction sequence (*interrupt-service subroutine*) from a specific memory location associated with the interrupt. The service routine can be a data-transfer routine after an ADC CONVERSION COMPLETE interrupt, an emergency-shutdown routine on power-supply failure, etc. As a service routine completes its job, it restores program counter and registers to return to the original program like any subroutine.

Interrupt operation would be simple if there were only one possible source of interrupts, but this is practically never true. Even a stand-alone digital computer usually has several interrupts corresponding to peripheral malfunctions (tape unit out of tape, printer out of paper), and flight simulators, space-vehicle controllers, and process-control systems may have hundreds of different interrupts.

A practical interrupt system, therefore, must:

1. "Trap" the program to different memory locations corresponding to specific individual interrupts, and
2. Deal with *priorities* in case of simultaneous or successive demands for service.

The most primitive system is to OR all interrupt flags onto a single interrupt line, whose interrupt-servicing routine then tests all the device flags in the sequence of their relative priority. As soon as the flag causing the interrupt is found, it will be necessary to disable transmission of all lower-priority flag levels. Such a system requires only simple electronics and disposes of the priority problem, but the flag-sensing program is time-consuming (n devices may require $\log_2 n$ successive decisions, even if the flag sensing is done by successive binary decisions). A somewhat faster method might be to employ a flag status word for indirect addressing.

A true *priority-interrupt system* involves a number of additional lines in the interface system. A device flag can cause an interrupt if (1) the interrupt is *enabled* (*armed*) by the digital-computer program, and (2) no higher-priority interrupt is being serviced.

Each interrupt-line subsystem, then, has three states: it may be inactive, enabled, or waiting for service. Figure 7-19 shows simplified hardware for one interrupt. An *interrupt request* (device-flag level) sets the *request flip-flop* in synchronism with a digital-computer-clock output (I/O sync), unless the interrupt is disabled by the program or by a higher-priority interrupt. The request flip-flop transmits an accurately timed interrupt-request step to the processor, which then disables all lower-priority interrupt lines and transmits an *interrupt-grant* level. The latter permits the processor-clocked *active flip-flop* to gate the starting address for the interrupt program into the memory-address register at the correct time in the computer cycle. The device flag is cleared at the same time. Different systems of priority-deciding logic exist, for instance:

1. The request flip-flops may be clocked in sequence of relative priority, so that the highest-priority request wins. This method is practical only with a relatively small number of devices, because it involves a delay in servicing lower-priority interrupts.
2. With interrupt circuits similar to that of Fig. 7-19 physically associated with each device, the devices are arranged on the party-line interrupt cable so that the device closest to the computer has the highest priority and can inhibit the request flip-flops of devices farther down the interrupt bus (PRIORITY IN and PRIORITY OUT terminations in Fig. 7-19).

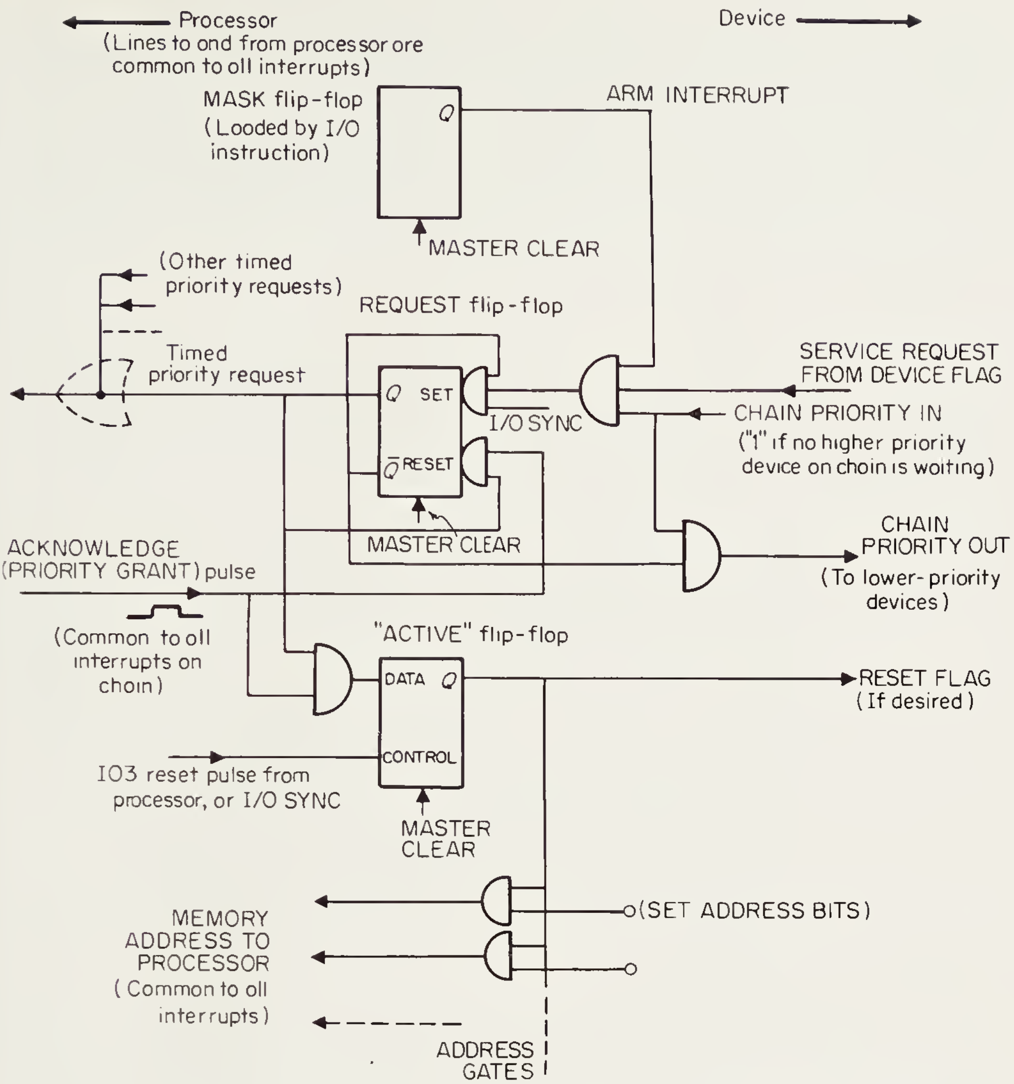

Processor
(Lines to and from processor are common to all interrupts)
Device
MASK flip-flop
(Loaded by I/O instruction)
Q
ARM INTERRUPT
MASTER CLEAR
(Other timed priority requests)
REQUEST flip-flop
Timed priority request
Q SET
$\bar{Q}$ RESET
I/O SYNC
SERVICE REQUEST FROM DEVICE FLAG
CHAIN PRIORITY IN
("1" if no higher priority device on chain is waiting)
MASTER CLEAR
ACKNOWLEDGE (PRIORITY GRANT) pulse
(Common to all interrupts on chain)
CHAIN PRIORITY OUT
(To lower-priority devices)
"ACTIVE" flip-flop
DATA Q
CONTROL
RESET FLAG
(If desired)
IO3 reset pulse from processor, or I/O SYNC
MASTER CLEAR
MEMORY ADDRESS TO PROCESSOR
(Common to all interrupts)
(SET ADDRESS BITS)
ADDRESS GATES

Fig. 7-19. Interrupt request/grant logic for one device. Similar logic can control direct memory access. (*From G. A. Korn, Minicomputers for Scientists and Engineers, McGraw-Hill, New York*, 1973.)

Substantially more complex interrupt logic exists. It may be possible to arm or disarm individual interrupts or groups of interrupts under the control of the main digital-computer program or by the interrupt-servicing routines. In general, the program will revert to the next-lower-priority interrupt-servicing routine and then to the main program after a high-priority interrupt has been serviced. Interrupts thus await service in a queue.

7-12. Direct Memory Access and Block Transfers. (a) Cycle Stealing.[50] Step-by-step program-controlled data transfers limit data-transmission rates and use valuable processor time for alternate instruction fetches and execution; programming is also tedious. It is often preferable to use additional hardware for interfacing a parallel data bus directly with the digital-computer memory data register and to request and grant one-cycle pauses in processor operation for direct transfer of data to or from memory (*interface* or *cycle-stealing* operation). In larger digital computers, a data bus may even access one memory bank without stopping processor interaction with other memory banks. To make direct memory access practical, we must be able to:

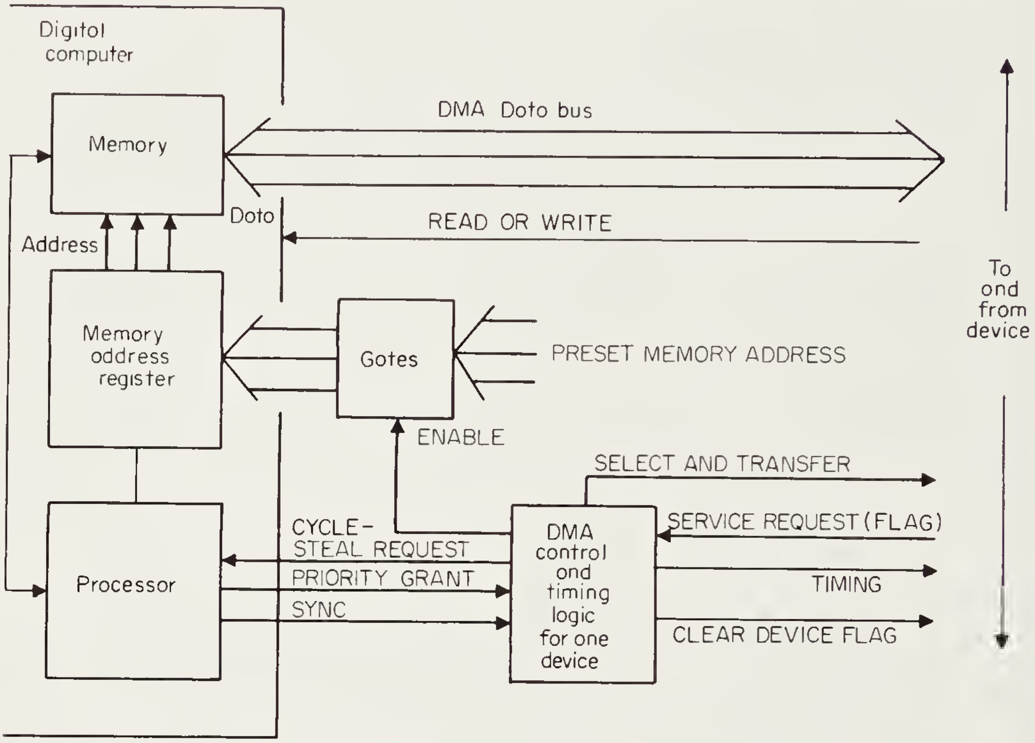


Fig. 7-20. A direct-memory-access (DMA) interface.

1. Address desired locations in memory
2. Synchronize cycle stealing with processor operation
3. Initiate transfers by device requests (this includes clock-timed transfers) or by the computer program

Since two or more devices may request data transfers at the same time, one requires *priority logic* essentially similar to that shown in Fig. 7-19. Figure 7-20 shows a simple *direct-memory-access* (DMA) *interface*. A *service request* (caused by device-flag level), unless inhibited by a higher-priority request, is answered by a *priority-grant* level. This signal sets a processor-clocked *enable flip-flop*, which strobes a suitable *memory address* into the processor memory-address register and then causes memory and device logic to transfer data from or to the DMA data bus.

In some computer systems (e.g., Digital Equipment Corporation PDP-15), the DMA data lines are identical with the programmed-transfer data lines. This simplifies interconnections at the expense of processor hardware. In other systems, the DMA data lines are used to transmit the DMA address to the processor before data are transferred. This further reduces the number of bus lines but complicates hardware and timing.

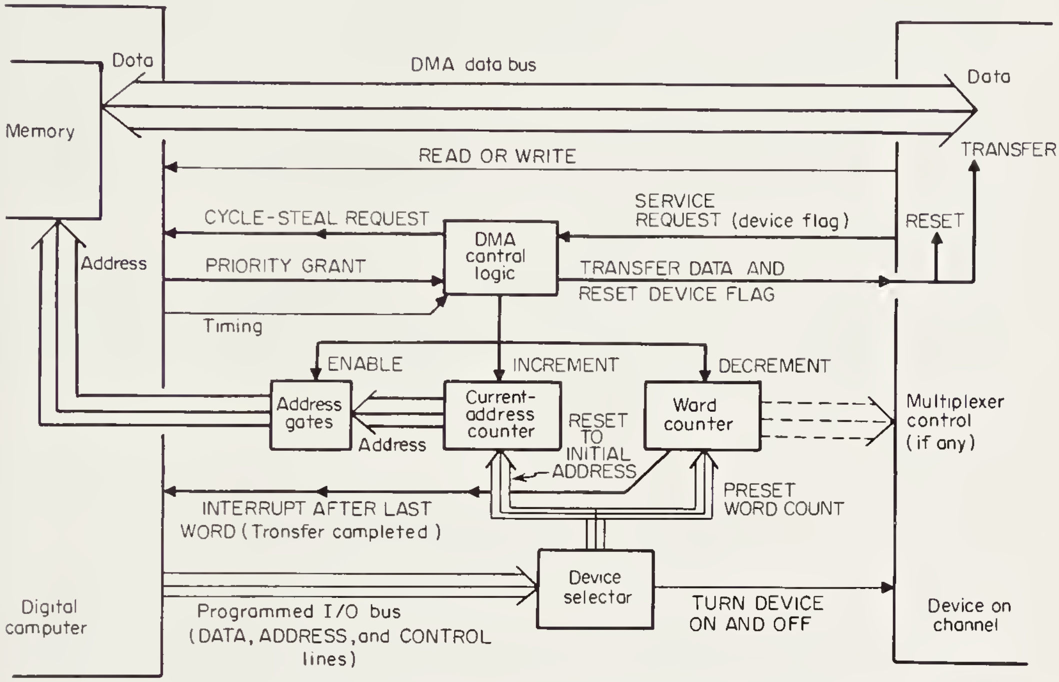


Fig. 7-21. A simple data channel for automatic block transfers. (*From G. A. Korn, Minicomputers for Scientists and Engineers, McGraw-Hill, New York*, 1973.)

Note that *cycle stealing in no way disturbs the program sequence*. Even though smaller digital computers must stop computation during memory transfers, the program simply skips a cycle at the end of the memory cycle (no need to complete the current instruction) and later resumes just where it left off. One does not have to save register contents or other information, as with program interrupts.

(b) Automatic Block Transfers.[50] As described, the DMA data transfer was *device*-initiated. A *program-dependent* decision to transfer data, even directly from or to memory, still requires a programmed instruction to cause a DMA service request. This is hardly worth the trouble for a *single-word* transfer. Most DMA transfers, whether device- or program-initiated, involve not single words but *blocks* of tens, hundreds, or even thousands of data words.

Figure 7-21 shows how the simple DMA system of Fig. 7-20 can be expanded into an *automatic data channel* for block transfers. Data for a block can arrive or depart asynchronously, and the DMA controller will steal cycles as needed and permit the program to go on between cycles. A block of words to be transferred will, in general, occupy a corresponding block of adjacent memory registers. Successive memory addresses can be gated into the memory-address register by a counter, the *current-address counter*. Before any data transfer takes place, a programmed instruction sets the current-address counter to a desired *initial address;* the desired number of words (*block length*) is set into a second counter, the *word counter*, which will count down with each data transfer until zero is reached after the desired number of transfers. As service requests arrive from, say, an analog-to-digital converter or data link, the DMA control logic implements successive cycle-steal requests and gates successive current addresses into the memory-address register as the current-address counter counts up.

The word counter is similarly decremented once per data word. When a block transfer is completed, the word counter may cause an interrupt, so that a new block of data can be processed. If desired, the word counter can also serve for sequencing device functions (e.g., for selecting successive ADC multiplexer addresses).

NOISE, GROUND SYSTEMS, AND INTERCONNECTIONS

7-13. Noise and Ground Systems. Noise in analog-computer signal circuits and ground systems is due to line-frequency components and harmonics, chopper noise, and/or high-frequency pickup from digital circuits and radio-frequency disturbances. A few millivolts rms of noise are usually tolerable. Fluctuations of millivolt amplitude will not affect solutions appreciably, but noise rectified in multipliers or function generators might be integrated to cause more serious errors. Chopper-frequency-noise pickup can cause chopper-channel offsets (Sec. 3-36).

Ground-system noise and common-ground impedances can cause serious problems, especially in low-voltage, high-frequency computers. A good earth ground is not always easy to come by, and the power-line "industrial" ground should be used for a-c return only. To minimize common-ground impedances, we select a single *common-ground point*, usually near a patchbay. We employ *separate* copper-bar, No. 10 wire or heavy braid returns from amplifier signal ground and/or chopper ground (Fig. 3-36), power ground, coefficient-potentiometer ground points for (small) groups of computing elements, and a single chassis-ground connection to the computer cabinet (Fig. 7-22).[52–56]

Even with a properly designed ground system, we still have electromagnetic noise pickup from ground loops, which are unavoidable at

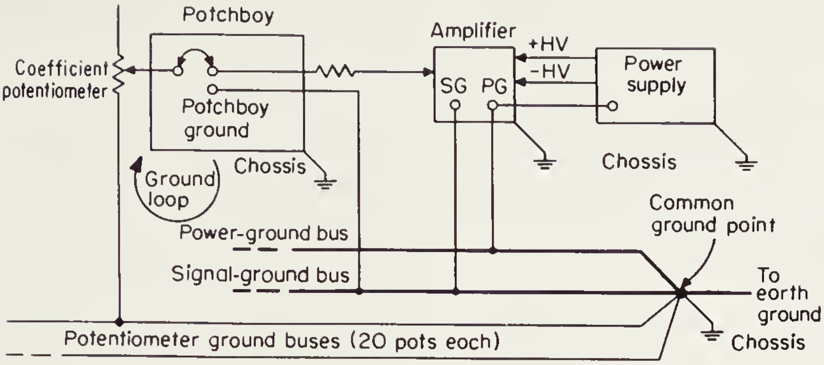


Fig. 7-22. Analog-computer ground system, showing separate ground returns and an unavoidable ground loop. Low-impedance transistor computers may require individual potentiometer returns.

least in general-purpose analog computers (Fig. 7-22). Circulating currents can cause ground-return voltage drops of the order of millivolts rms. Aside from high-frequency pickup, ground noise is mostly induced by magnetic fields at the line frequency and its harmonics. Third-harmonic pickup can be especially troublesome but may be reduced by the symmetrical input-transformer connection of Fig. 7-23, and possibly by power-line filters and balancing. Harmonic fields may also be generated by saturable-reactor filament-voltage regulators and by ordinary transformers operated near their maximum ratings. If practicable, the entire power-supply system can be housed away from the main computer cabinet; but old vacuum-tube computers still employ individual filament transformers on the computing-element modules to avoid high-current a-c wiring. All a-c filament leads must be twisted.

Electrostatic shielding of signal wiring, modules, and power-supply transformers reduces high-frequency pickup but increases shunt capacitances. In stubborn cases, even ground returns and filament leads have been shielded, while "slow" analog computers may omit shielding of low-impedance amplifier-output connections. Signal-cable shields must never carry ground-return currents; a ground-return line can be twisted with the signal lead inside a shield grounded at the source end. A system earth ground is used for electrical safety but may actually increase high-frequency noise[55] because the ground lead can act as an antenna! It may pay

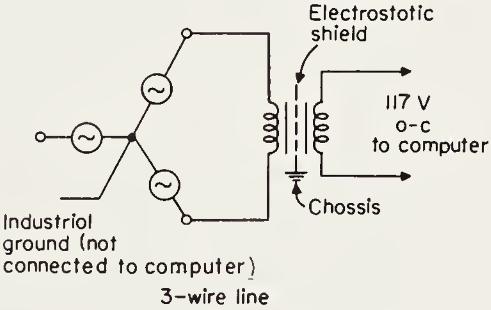


Fig. 7-23. Use of an isolation transformer to cancel third-harmonic line-voltage components. (*Contributed by H. Stewart, University of Arizona.*)

to shield the ground lead with copper tubing grounded at an experimentally determined point or points.[52]

7-14. Digital-system Noise. Interconnection Problems. Digital-system noise, especially high-frequency spikes and pulse ringing, can cause very objectionable noise in wideband analog circuits via ground currents and radiation. This can be true even though the digital circuits themselves work well within their noise-immunity limits. Transistor/transistor logic (TTL) and diode/transistor logic (DTL), with their harsh ground-current transients and relatively high output impedances, are bad offenders in this respect. *Emitter-coupled logic* (ECL) *has near-constant ground current and low output impedances and is recommended as the best choice for critical wideband hybrid computing circuits.*[14,70] The following general rules apply:

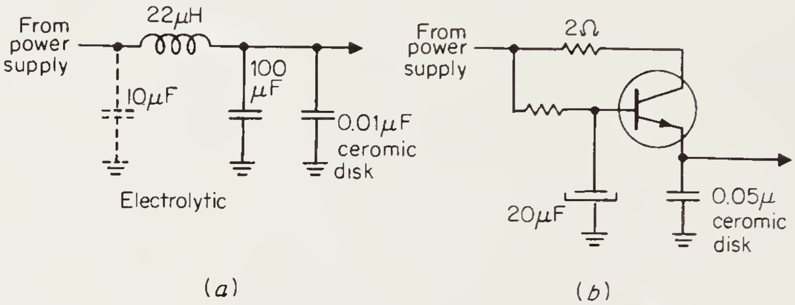


Fig. 7-24. Decoupling circuits. The ceramic-disk capacitor in (*a*) "tunes out" the inductance of the electrolytic capacitor. Every power-supply line on every card should have a decoupling circuit.

1. *To reduce the digital noise at its source*, avoid long lines which are not carefully terminated and/or clamped to prevent ringing (consult the applications manual for the type of logic circuits used). Wires from each circuit should fan out, not continue from point to point.
2. *It is important to keep as much of the digital noise as possible within the digital-circuit cards and out of power-supply and ground circuits and to keep the remainder out of the analog circuits as much as possible.* For this reason, every digital and analog circuit card should have a *decoupling circuit* on each power-supply line (Fig. 7-24). A *ground plane* ought to be used on each card, not so much for electrostatic shielding as to reduce the areas of inductive loops. *Laminated power-supply buses* also help with decoupling.
3. Digital ground returns (which act as transmitting antennas!) should be kept separate and well away from analog ground leads. It is, in fact, a good idea to keep digital and analog circuits in separate shielded enclosures and also to operate their power

supplies from different phases of the three-phase line, with *RF* filters in each a-c lead.

4. *Where a digital computer is connected to a linkage or analog computer, we must avoid returning any digital signals through the common ground.* It is best to use push-pull line drivers with twisted and shielded lines into differential line receivers or at least differential line receivers fed with the digital signal and digital ground-return lines. Balun transformers have also been successfully used to replace actual differential lines.[74]

   Very slow logic (below 50 to 100 KHz) can be transmitted from a digital controller to an analog computer through light-coupled switches (Fig. 5-6*d*), but this is not practical for general-purpose hybrid computation.
5. Finally, make sure that all operational amplifiers and other analog feedback circuits are equalized so as to avoid high-frequency peaks (which cause ringing when excited by digital noise).

With all these precautions, it is possible to keep digital noise on analog circuits below 5 to 10 mV peak-to-peak.

REFERENCES AND BIBLIOGRAPHY

Analog/Hybrid Computer Systems

(See also *Proc. 1st to 6th AICA Conf.*, Presses Académiques Européennes, Brussels)

1. Kusto, T. J.: Patchbays, Interconnections, and Enclosures, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
2. Loveman, B. D.: Semiautomatic Programming, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
3. ———: Problem-checking Circuits and Procedures, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
4. Rogers, S.: Complete Electronic Analog Computers: Large Installations, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
5. Meyer-Brötz, G.: Ein transistorisierter Präzisions-Analogrechner, *Telefunken Z.*, September 1960.
6. Eckes, H. R., and G. A. Korn: Digital Program Control for Iterative Differential Analyzers, *ACL Memo* 86, Electrical Engineering Department, University of Arizona, 1963; *Simulation*, February 1964.
7. Korn, G. A.: High-speed Iterative Analog-digital Computer: The ASTRAC II Project, *Proc. 4th AICA Conf.*, Brighton, England, 1964; Presses Académiques Européennes, Brussels, 1965.
8. Survey of Commercially Available Hybrid-computer Systems (Seven invited papers), *Simulation*, January, February, April, June, July, August, and October 1965.
9. Giloi, W., et al.: Aufbau und Anwendung des hybriden Analogrechners, *Telefunken Z.*, January 1966 (see also the descriptions of Telefunken hybrid computers in the same issue).
10. Korn, G. A.: APE II: A Modern Analog/Hybrid Computer for Laboratory Instruction, *Simulation*, February 1967.
11. ———: New Packaging Techniques for Integrated-circuit Hybrid Computers: The APE II and LOCUST Systems, *Proc. 5th AICA Conf.*, Lausanne, 1967; Presses Académiques Européennes, Brussels, 1968.
12. ———: The SUPERAPE Concept, *Simulation*, April 1968.

13. Conant, B. K.: Modern Digital Control for a Small Iterative Analog Computer, *Ann. AICA*, January 1968.
14. ———: A New Iterative Differential Analyzer, *Proc. FJCC*, 1968.
15. Keene, D. H.: The Effect of IC's on the Design of Analog Computers, *Simulation*, April 1969.

Automatic Analog-computer Patching and Programming

(See also Refs. 79 and 80)

16. Kamm, L. J.: Analog-computer Problem Connector Employs Punched Card and Crossbar, *Electronics*, Aug. 18, 1961.
17. ———: Contact Pattern Aligns Printed-circuit Matrix, *Electronics*, Aug. 25, 1961.
18. Green, C., et al.: The Code APACHE for the Programming of an Analog Problem by Means of a Digital Computer, *Proc. 3d AICA Conf.*, Opatija, Yugoslavia, 1961, Presses Académiques Européennes, Brussels, 1962; reprinted in *Ann. AICA*, April 1963.
19. ———: APACHE: A Breakthrough in Analog Computing, *IRETEC*, October 1962.
20. Stein, M. L.: Automatic Digital Programming of Analog Computers, *IEEETEC*, April 1963.
21. Burgin, G. H.: APACHE: Some Encouraging Experiences, *Simulation*, January 1966.
22. De Backer, W., and A. Van Wauwe: The SIOUX System and Hybrid Block Diagrams, *Simulation*, July 1965.
23. Starr, D. A., and J. J. Jonsson: The Design of an Automatic Patching System, *Simulation*, June 1968.
24. Hannauer, G.: Automatic Patching for Analog and Hybrid Computers, *Simulation*, May 1969.
25. Gracon, T. J., and J. C. Strauss: Automatic Analog-computer Patching Systems, *Simulation*, September 1969.
26. ———: Design of an Automatic Patching System for Analog Computers, *Proc. SJCC*, 1970.
27. Howe, R. M., et al.: Electronic Patching for the AD-4 Hybrid Computer, *Proc. FJCC*, 1970; see also *Proc. Summer Computer Simulation Conf.*, Denver, Colo., 1970.
28. Yagi, A.: A Solid-state Crossbar Switch for Automatic Analog-computer Patching, *Simulation*, January 1972.
29. Sage, M. W., et al.: Analog Programming by Digital Computers (APSE), *Proc. 5th AICA Conf.*, Lausanne, 1967.
30. Rook, P. E.: ACTRAN: An Analog/Hybrid Compiler, *Proc. 6th AICA Conf.*, Munich, 1970.
31. Kurokawa, K.: All-IC Hybrid Computer Eliminates Patchwork, *Electronics*, Mar. 17, 1969; see also *Proc. 6th AICA Conf.*, Munich, 1970, and Ref. 79.

Analog-computer Test Circuits and Maintenance Procedures

(See bibliography for Chap. 9 for automatic hybrid-computer checkout programs)

32. Hamer, H., and J. D. Kennedy: Testing Operational Amplifiers, *J. Assoc. Computing Machinery*, April 1955.
33. Horwitz, R. D.: Maintaining an Analog Computing System, *Electron. Inds.*, December 1958.
34. Sykes, R. P.: A Preventive-maintenance Program for Large General-purpose Electronic Analog Computers, *Memorandum*, Space Technology Laboratories (then Ramo-Wooldridge Corp.), Los Angeles, Calif., 1958.
35. ———: Calibration of High-resistance Padded Resistors, *Instr. Control Systems*, October 1959.
36. ———: Calibrating Analog-computer Integrators, *Instr. Control Systems*, September 1960.
37. Ehlers, H. L., and W. Hochwald: Maintenance Procedures and Test Circuits, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
38. Loveman, B. D.: Problem-checking Circuits and Procedures, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.

Direct-writing Recorders

39. Pierce, D. E.: Pressurized-ink Recording, *Instr. Control Systems*, March 1962.
40. Zusi, F. C.: Nomenclature for Direct-writing Recorders, *Instr. Control Systems*, July 1961.
41. Miller, A.: Factors Affecting Accuracy of Oscillographic Records, *The Right Angle*, Sanborn Co., Waltham, Mass., August and November 1953.
42. ———: Design Considerations of d'Arsonval-galvanometer/Power-amplifier Systems, *The Right Angle*, Sanborn Co., Waltham, Mass., August and November 1958.
43. Vance, P. R., and D. L. Haas: An Input-Output Unit for Analog Computers, *Proc. IRE*, October 1953.

Interface Design

(Refer also to the interface manuals of various digital-computer manufacturers)

44. Borger, E. R.: Characteristics of Priority Interrupts, *Datamation*, June 1965.
45. Andelman, S. J.: Real-time I/O Techniques, *Computer Design*, May 1966.
46. Klerer, M., and G. A. Korn: *Digital Computer User's Handbook*, McGraw-Hill, New York, 1967.
47. Schmidt, W. E.: Methods for Priority Interrupts and Their Implication for Hybrid Programs, *Simulation*, July 1967.
48. Pracht, C. P.: A New Digital-attenuator System, *Simulation*, April 1967.
49. Marston, G. P.: A Medium-scale Hybrid Interface, *Simulation*, May 1968.
50. Korn, G. A.: Digital-computer Interface Systems, *Simulation*, December 1968.
51. Wilkins, J.: "The PDP-9/LOCUST Interface," M.S. Thesis, University of Arizona, 1969.

Shielding, Interconnections, and Grounding

52. Stewart, E. L.: Grounds, Grounds, and More Grounds, *Simulation*, August 1965.
53. ———: Noise Reduction in Interconnect Lines, *Simulation*, September 1965.
54. Sargent, R. S., and T. R. Kuchlewski: Reduction of Noise in Low-impedance (Ground) Lines, *Analog Rept.* 16, Martin Co., Baltimore, Md., 1963.
55. Marsh, J. M., and V. P. Scott: Analog-facility Grounding Techniques, WAPD-T-1701, Bettis Atomic Power Laboratory, Westinghouse Electric Company, Pittsburgh, Pa., 1964.
56. Marimon, R. L.: Ground Rules for Low-frequency High-gain Amplifiers, *Electron. Design*, Apr. 12, 1963.
57. Morrison, R.: Shielding Signal Circuits, *ISA J.*, March 1966.
58. EID Staff: Grounding Low-level Instrumentation Systems, *EID*, January 1966.
59. Ginn, D.: Data-system Grounding Techniques, *Telemetry*, July 1966.
60. Burdi, A. J.: Keep Sampled-data Systems Accurate, *Electron. Design*, May 24, 1966.
61. McCullough, W.: How to Reduce Common-mode Effects, *EEE*, February 1967.
62. Bowers, W. J.: Minimizing Common-mode Noise, *Control Eng.*, July 1967.
63. Dynamic Bridge Differential Amplifier 770-440, Redcor Corporation, Canoga Park, Calif., 1967.
64. Klipec, B.: Controlling Noise in Instrument Circuits, *Control Eng.*, March 1968.
65. Bailey, S. J.: Using Telephone Lines for Critical Signal Transmission, *Control Eng.*, March 1969.
66. Buntenbach, R. W.: On the Design of Ground Circuits, STAR N68-3573, Lawrence Rad. Lab., Livermore, Calif., 1968.
67. Budzilovich, P. N.: Electrical Noise: Its Nature, Causes, Solutions, *Control Eng.*, May 1969.
68. Morrison, B.: *Grounding and Shielding Techniques in Instrumentation*, Wiley, New York, 1967.

Digital-circuit Noise

69. Jones, J. P.: *Causes and Cures of Noise in Digital Systems*, Computer Design Publication Company, West Concord, Mass., 1964.
70. Korn, G. A.: Reduction of Digital Noise in Hybrid Analog-Digital Computers, *Simulation*, March 1966.

71. Printed-circuit Shielding in AFAL-TR-66-371, Bunder-Ramo Corporation, April 1967.
72. Walker, R. M., and R. A. Aldrich: Standard IC's for Digital Data Communication, *The Electronic Engineer*, March 1968.
73. Saenz, R. G., and E. M. Fulcher: An Approach to Logic Circuit Noise, *Computer Design*, April 1969.
74. *Balun Applications*, Pulse Eng., Inc., Santa Clara, Calif., 1969.

Miscellaneous

75. Time-critical Simulation Systems, *General Information Manual* 44629200, Control Data Corp., Sunnyvale, Calif., 1970.
76. Gilbert, E. G., and R. A. Moran: A General Method for Programming Synchronous Logic in Analog Computation, *Proc. FJCC*, 1968.
77. Campbell, J. B.: The Economics of Scale in Hybrid Computing, *Proc. FJCC*, 1970.
78. Korn, G. A., et al.: A New Graphic-display/Plotter for Small Digital Computers, *Proc. SJCC*, 1969.
79. *Proc. AICA Conference on Simulation of Complex Systems* (contains several papers on automatic analog-computer patching), Society of Analog Technique, Tokyo, Japan, 1971.
80. Rigas, H. B., and D. J. Coombs: PATCH, Analog-computer Patching from a Digital Simulation Language, *IEEETC*, October 1971.

CHAPTER 8

SPECIAL COMPUTING TECHNIQUES

INTRODUCTION

8-1. Requirement for Advanced Computing Techniques. Analog computers, and especially fast analog computers, are, essentially, relatively inaccurate. Although static component accuracies within 0.1 percent of half-scale are not uncommon, dynamic errors measured at the 100-Hz to 100-KHz signal frequencies of modern iterative differential analyzers are rarely below 0.5 percent of half-scale. Errors in this range would be perfectly acceptable in most engineering applications if we could only be sure that component errors will not accumulate into intolerable solution errors.

A very powerful computing technique employs the analog computer to produce *corrections* to an accurate analytically or digitally precomputed approximate solution (Secs. 8-1 to 8-4). Such a *perturbation method* may reduce solution errors well below the level of analog-component errors, which will now affect only a small correction to our solution. Where applicable, such techniques are greatly preferable to expensive and probably futile schemes for increasing analog component accuracies. New combinations of analog and digital computing elements often make it possible to let the accurate but slow digital computer produce a precisely known drift- and noise-free approximate solution, which can then be corrected by fast analog computing elements.

We are, next, vitally interested in *methods for estimating component-error effects on computer solutions.* Sections 8-4 and 8-5 discuss techniques employing the computer itself to perform such estimates.

The development of new fast iterative differential analyzers with memory and sophisticated sequential logic (Chap. 7) offers computing possibilities way beyond ordinary direct simulation, but only if mathematical development of convergent iteration techniques and error control keeps pace with circuit and system design. The most urgent requirement is for methods utilizing the extraordinary speed of iterative analog computation directly to make up for its poor accuracy. Aside from random-process studies,[132] the most promising application of iterative differential analyzers is *automatic parameter optimization* (Secs. 8-7 to 8-16); many important problems can be reduced to the computation of maxima or minima. Even more badly needed, but in a less satisfactory state of development, are computer techniques for *iterative solution of partial differential equations*, especially in connection with process-control and nuclear-power applications (Refs. 87 to 108).

GAUGING AND IMPROVING COMPUTER ACCURACY: PERTURBATIONS, PARAMETER SENSITIVITY, RESUBSTITUTION, AND AUTOMATIC SCALING

8-2. Perturbations and Linearization. Our problem is to solve the first-order differential equation

$$\frac{d}{dt}x(t) = f(t,x;\alpha) \tag{8-1}$$

for the state variable $x = x(t)$, where α is a given parameter, initial value, or forcing function; note that α can be a function $\alpha(t)$ of the independent variable t. We assume that $\alpha(t) = \alpha_0(t) + \delta\alpha(t)$ does not change greatly from its nominal or unperturbed value $\alpha_0(t)$, and that we know the *nominal or unperturbed solution* $x_0(t)$ defined by

$$\frac{d}{dt}x_0(t) = f(t,x_0;\alpha_0) \tag{8-2}$$

accurately from analysis or from another computation. Granting suitable differentiability, we introduce $\delta x = x - x_0 = \delta x(t)$ and use the Taylor-series approximation

$$\frac{dx}{dt} = \frac{dx_0}{dt} + \frac{d}{dt}\,\delta x = f(t,x;\alpha) \approx f(t,x_0;\alpha_0) + f_x(t,x_0;\alpha_0)\,\delta x + f_\alpha(t,x_0;\alpha_0)\,\delta\alpha \tag{8-3}$$

We subtract Eq. (2), so that, *for sufficiently small* $\delta\alpha$, δx,

$$\frac{d}{dt}\,\delta x = f_x(t,x_0;\alpha_0)\,\delta x + f_\alpha(t,x_0;\alpha_0)\,\delta\alpha \tag{8-4}$$

This *perturbation equation** is a *linear* differential equation relating the unknown small *perturbation* δx of our nominal solution to the given input $\delta\alpha$. Computer solution of Eq. (4) for δx is simple *if* we can produce the derivatives $f_x(t,x_0;\alpha_0), f_\alpha(t,x_0,\alpha_0)$ which will, in general, depend on the nominal solution $x_0(t)$. *If $x_0(t)$ is accurately precomputed analytically, digitally, or by accurate "slow" analog computation, then errors in the solution $\delta x(t)$ of Eq. (4) will have a relatively small effect on the total solution $x(t) = x_0(t) + \delta x(t)$.*

Our linear perturbation analysis is easily extended to cases involving n state variables $x_1, x_2, \ldots, x_n$ and r parameters $\alpha_1, \alpha_2, \ldots, \alpha_r$. Given

$$\left.\begin{aligned}&\frac{dx_i}{dt} = f_i(t;x_1,x_2,\ldots,x_n;\alpha_1,\alpha_2,\ldots,\alpha_r)\\ &\text{with } x_i = x_{0i} + \delta x_i \qquad \alpha_k = \alpha_{0k} + \delta\alpha_k\end{aligned}\right\} \begin{aligned}(i &= 1,2,\ldots,n;\\ k &= 1,2,\ldots,r)\end{aligned} \qquad (8\text{-}5)$$

we find, assuming differentiability and valid linearization,

$$\frac{d}{dt}\,\delta x_i = \sum_{k=1}^{n} \frac{\partial f_i}{\partial x_k}\,\delta x_k + \sum_{k=1}^{r} \frac{\partial f_i}{\partial \alpha_k}\,\delta\alpha_k \qquad (i = 1,2,\ldots,n) \qquad (8\text{-}6)$$

This is again, a linear system with (in general) variable coefficients. Equations (1) and (4) are equivalent to Eqs. (5) and (6) if we regard $x, x_0, \delta x, f$ as $n \times 1$ matrices, $\alpha, \alpha_0, \delta\alpha$ as $r \times 1$ matrices, f_x as an $n \times n$ matrix, and f_α as an $n \times r$ matrix.

The linear nature of the perturbation equations (4) or (6) can be of the greatest value even where increased accuracy is not needed. In an important class of applications (e.g., space-vehicle trajectory computation), $\delta\alpha$ represents a small random forcing function or random initial-condition perturbation with known statistics. It is then relatively simple to obtain the resulting mean-square perturbation of x or x_i.[3.132]

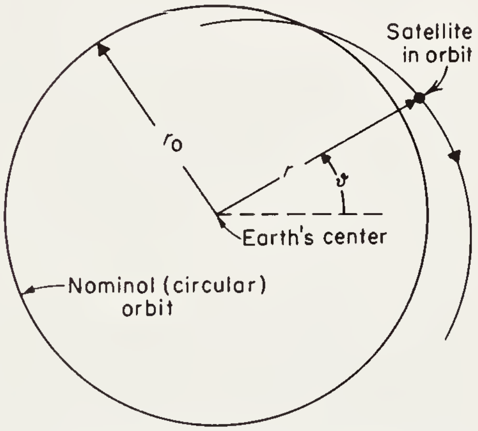


Fig. 8-1. Satellite-orbit geometry.

8-3. Example: Earth-satellite Orbit and Reentry Computation.[8–10]

Figure 8-1 illustrates a plane earth-satellite trajectory. In terms of the inertial-reference coordinates r, ϑ, the equations of motion are

$$m\frac{d\dot{r}}{dt} = mr\dot{\vartheta}^2 - \frac{mg_0r_0^2}{r^2} + F_r \qquad m\frac{d}{dt}(r^2\dot{\vartheta}) = rF_\vartheta \qquad (8\text{-}7)$$

if we regard the vehicle as a constant point mass. $mr\dot{\vartheta}^2$ is a kinetic reaction (centrifugal "force"), $mg_0r_0^2/r^2$ is the idealized inverse-square-law gravity force, and F_r, F_ϑ are forces due to aerodynamic effects and gravity changes.

---

* See footnote to Sec. 8-4.

For an ideal circular orbit with $F_r = F_\vartheta = 0$, $r = r_0$ implies

$$\vartheta = \vartheta_0(t) = \vartheta(0) + t\sqrt{\frac{g_0}{r_0}} \tag{8-8}$$

Since the subtraction of the two large first terms on the right of the first Eq. (7) tends to make direct computer solutions inaccurate, we write

$$r = r_0(1 + \delta\rho) \qquad r\dot{\vartheta} = \sqrt{g_0r_0}\,(1 + \delta u) \tag{8-9}$$

$\delta\rho$ and δu are new dimensionless variables describing the relative perturbation of a nominal circular orbit due to nonzero values of F_r, F_ϑ, $\dot{r}(0)$, and $\dot{\vartheta}(0) - \sqrt{g_0/r_0}$. $\delta\rho$ measures the relative altitude increase, and δu is the fractional change in the horizontal velocity component. We introduce Eqs. (8) and (9) into Eq. (7) to find new equations of motion in terms of $\delta\rho$ and δu:

$$\left.\begin{aligned} \frac{d^2}{dt^2}\,\delta\rho &= \frac{g_0}{r_0}\left[\frac{2\delta u + \delta u^2}{1 + \delta\rho} + \frac{\delta\rho}{(1 + \delta\rho)^2} + \frac{F_r}{mg_0}\right] \\ \delta u &= \frac{1}{1 + \delta\rho}\left\{\sqrt{\frac{g_0}{r_0}}\int_0^t (1 + \delta\rho)\,\frac{F_\vartheta}{mg_0}\,dt - \delta\rho\right. \\ &\qquad\qquad \left. + (\delta\rho + \delta u + \delta\rho\;\delta u)_{t=0}\right\} \end{aligned}\right\} \tag{8-10}$$

Note carefully that the differential equations (10) are *not* linearized; their exact solution would yield the perturbations $\delta\rho$, δu *exactly*. In practice, computer errors affect only perturbations of a precisely (analytically) known nominal orbit, so that remarkably accurate trajectory computations result. For $r_0 = 4{,}000$ miles, $r_0\,|\delta\rho| < 80$ miles, we have $|\delta\rho| < 0.02$, so that $5{,}000\,\delta\rho$ is a suitably scaled computer voltage. Fogarty and Howe[8-10] have implemented the divisions by $1 + \delta\rho$ required for Eq. (10) so that the effects of multiplication errors on the quotient were relatively small. Errors in many computed earth orbits were within 200 ft.

If the relative perturbations $\delta\rho$ and δu remain small compared with unity, one can linearize the equations of motion (10) by neglecting nonlinear terms. In particular, for the classical case of a low-eccentricity elliptical inverse-square-law orbit in air-free space, Eq. (10) reduces approximately to

$$\frac{d^2}{dt^2}\,\delta\rho = \frac{g_0}{r_0}(2\delta u + \delta\rho) \qquad \delta u = -\delta\rho + [\delta\rho + \delta u]_{t=0} \tag{8-11}$$

which correctly describes sinusoidal perturbation of a nominal circular orbit by given initial velocities.

8-4. Parameter-influence Coefficients and Parameter-sensitivity Studies.[1-7] If the parameter α in the given differential equation (1) is independent of t (e.g., a constant coefficient or an initial value), we can define the *parameter-influence coefficient*

$$\lim_{\delta\alpha\to 0} \frac{\delta x(t)}{\delta\alpha} = \frac{\partial x}{\delta\alpha} = x_\alpha(t) \tag{8-12}$$

Again, assuming suitable differentiability, $x_\alpha(t)$ *measures the sensitivity of our solution* $x(t)$ *to small changes* $\delta\alpha$ *in the parameter* α. For each nominal parameter value $\alpha = \alpha_0$,

$$x(t) = x_0(t) + \delta x \approx x_0(t) + x_\alpha(t)\,\delta\alpha \tag{8-13}$$

The parameter-influence coefficient $x_\alpha(t)$ satisfies the linear *sensitivity equation**

$$\frac{d}{dt} x_\alpha(t) = f_x(t,x_0;\alpha_0)x_\alpha(t) + f_\alpha(t,x_0;\alpha_0) \tag{8-14}$$

obtained by partial differentiation of the given differential equation (1) with respect to α. We can obtain $x_\alpha(t)$ as a solution of Eq. (14) with the suitably assigned initial value $x_\alpha(0)$.

In the multidimensional case, we have an $n \times r$ matrix of parameter-influence coefficients $x_{ik}(t) = \partial x_i/\partial\alpha_k$ $(i = 1, 2, \ldots, n; k = 1, 2, \ldots, r)$, with

$$x_i(t) = x_{0i}(t) + \delta x_i \approx x_{0i}(t) + \sum_{k=1}^{r} x_{ik}(t)\,\delta\alpha_k \qquad (i = 1, 2, \ldots, n) \tag{8-15}$$

and $n \times r$ linear sensitivity equations (see also Sec. 8-14*c*)

$$\frac{d}{dt} x_{ik}(t) = \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} x_{jk}(t) + \frac{\partial f_i}{\partial \alpha_k} \qquad (i = 1, 2, \ldots, n; k = 1, 2, \ldots, r) \tag{8-16}$$

Analytical or computer studies of parameter sensitivity can yield valuable information on the effects of various computer errors[1-4] and on the variation of solutions over reasonable parameter ranges. With modern iterative differential analyzers, however, there is less need for actual computer solution of sensitivity equations. The true effects of parameter changes switched in and out during alternate repetitive-computer runs are conveniently observed on an oscilloscope without any need for linearization.[11] If function memory is available, we can also compute true perturbations $\delta x(t)$ by subtracting the stored nominal solution $x_0(t)$ from the perturbed solution $(x)t$ or by correlation techniques.[15,16]

---

* While $\delta x(t)$ satisfies Eq. (4) only approximately for small $\delta\alpha$, $x_\alpha(t)$ satisfies Eq.(14) *exactly*. $x_\alpha(t)\,\delta\alpha = \delta^1 x(t)$, which satisfies the linear perturbation equation exactly, is a *first-order perturbation* approximating $\delta x(t)$ for small $\delta\alpha$, δx.

8-5. Resubstitution Checks. Exact solutions $x_i = x_i(t)$ of a given system of differential equations

$$\frac{dx_i}{dt} = f_i(t;x_1,x_2, \ldots ,x_n) \qquad (i = 1,2, \ldots ,n) \tag{8-17}$$

must satisfy

$$x_i(t) = \int_0^t f_i(t;x_1,x_2, \ldots ,x_n)\, dt + x_i(0) \qquad (i = 1,2, \ldots ,n) \tag{8-18}$$

If now the $x_i(t)$ are replaced by necessarily nonideal computer solutions $x_i'(t)$, then the functions

$$\Delta x_i(t) = \int_0^t f_i(t;x_1',x_2', \ldots ,x_n')\, dt + x_i'(0) - x_i'(t) \qquad (i = 1,2, \ldots ,n) \tag{8-19}$$

are measures of the solution errors. Each $\Delta x_i(t)$ is, in fact, a useful correction whose addition to $x_i'(t)$ tends to improve our computed solution.[12,13] This fact becomes interesting when we can implement the checking operation (19) economically and more accurately than our original differential-equation-solving operations. In particular, a repetitive analog computer with scanning readout (Sec. 7-3) can read solutions into accurate "slow" analog computing elements or into a digital computer to check selected solutions. Correction of fast solutions would require function memory, although partial correction may be simpler. The validity of corrections (19) computed with *imperfect* computing equipment is an interesting topic for further investigation.[12]

8-6. Special Scaling Problems and Automatic Scaling.[14,19,20] When the magnitude of a machine problem variable varies over a wide range in the course of a problem solution, automatic rescaling at appropriate points of the computer setup may yield significantly increased solution accuracy. Examples are found in nuclear-reactor simulation and in control problems where an initially large error is nulled by the control-system action, as in homing-missile simulation.

Figure 8-2*a* illustrates the use of comparator/switch circuits for automatic rescaling of the divisor Y in an electronic divider. In some problems, we shall require dual comparators for testing $|Y|$ rather than

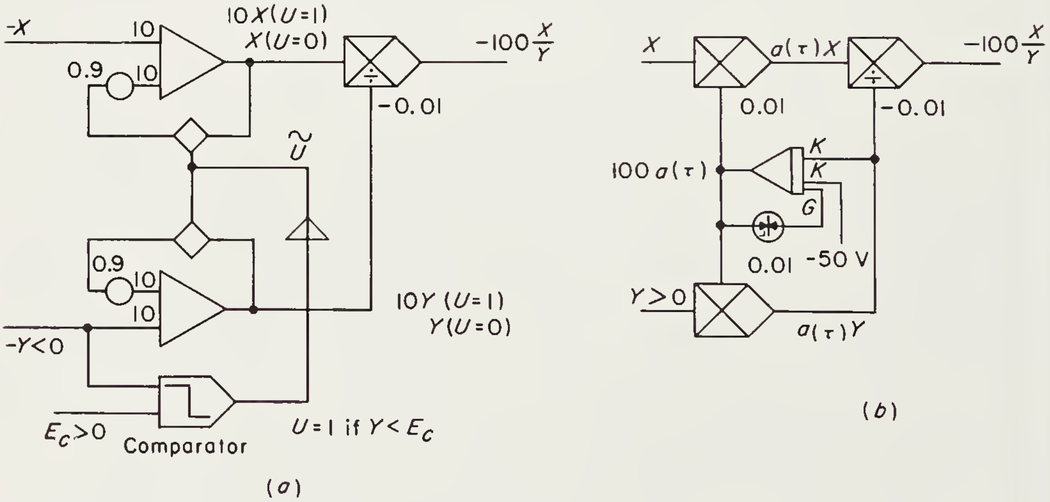


Fig. 8-2. Automatic scaling for a divider circuit. The comparator in (*a*) need not be accurate.

simply Y as in Fig. 8-2*a*, but relatively inaccurate comparators, such as the inexpensive circuit of Fig. 4-18*a*, are perfectly sufficient for automatic scaling. Figure 8-2*a* shows a circuit for *continuous* automatic rescaling.[131] True automatic rescaling can sometimes be replaced by scale-factor changes programmed as functions of time.

In many control problems, an error measure $v(t)$ is always small but appears as the difference of two large quantities. We can sometimes avoid the resulting large percentage errors by computing $v(t)$ as $\int_0^T \dot{v}(t)\,dt + v(0)$ at the expense of additional equipment. In beam-rider missile control systems, for instance, a problem variable

$$v = -x \sin \vartheta + y \cos \vartheta \tag{8-20}$$

is a small difference, while

$$\dot{v} = -\dot{x} \sin \vartheta + \dot{y} \cos \vartheta - (x \cos \vartheta + y \sin \vartheta)\dot{\vartheta} \tag{8-21}$$

has a magnitude comparable with those of the various terms on the right.[131]

CONTINUOUS OPTIMIZATION, EQUATION SOLVERS, CONSTRAINTS, AND MODEL MATCHING

8-7. Maxima and Minima. (a) Introduction. Many practical and theoretical problems require determination of parameter combinations $(x_1, x_2, \ldots, x_n)$ which will maximize or minimize a suitably given function

$$F = F(x_1, x_2, \ldots, x_n) \tag{8-22}$$

(*criterion function; objective function, measure of effectiveness*), possibly subject to constraints of the form

$$\varphi_k(x_1, x_2, \ldots, x_n) = 0 \qquad (k = 1, 2, \ldots, m) \tag{8-23}$$

and/or

$$\Psi_k(x_1, x_2, \ldots, x_n) \geq 0 \qquad (k = 1, 2, \ldots, r) \tag{8-24}$$

Practically all engineering problems involve parameter optimization (maximum performance, minimum cost), and a much larger class of problems can be reduced to optimization by way of minimum-error requirements. Error minimization through stable feedback-loop setups is a natural application for conventional and iterative analog computers, but also for digital and hybrid machines. In the following, we shall assume the existence of meaningful criterion functions and constraints, all differentiable as needed; and we assume the actual existence of maxima or minima, each with a finite neighborhood without other maxima or minima.

The simplest optimizer setup on an analog computer is merely a computing-element block generating $F(x_1, x_2, \ldots, x_n)$, with each parameter x_i adjusted manually to maximize or minimize the output (optimization by trial and error, Fig. 8-3a). To account for constraints (23) on the variables x_i, we simply optimize the modified criterion function

$$\Phi(x_1,x_2,\ldots,x_n) \equiv F(x_1,x_2,\ldots,x_n) + \tfrac{1}{2}\sum_{k=1}^{m} c_k\,\varphi_k^2(x_1,x_2,\ldots,x_n) \qquad (8\text{-}25)$$

or

$$\Phi(x_1,x_2,\ldots,x_n) \equiv F(x_1,x_2,\ldots,x_n) + \sum_{k=1}^{m} c_k\,|\varphi_k(x_1,x_2,\ldots,x_n)| \qquad (8\text{-}26)$$

where the c_k are arbitrary but large positive constants for minimization

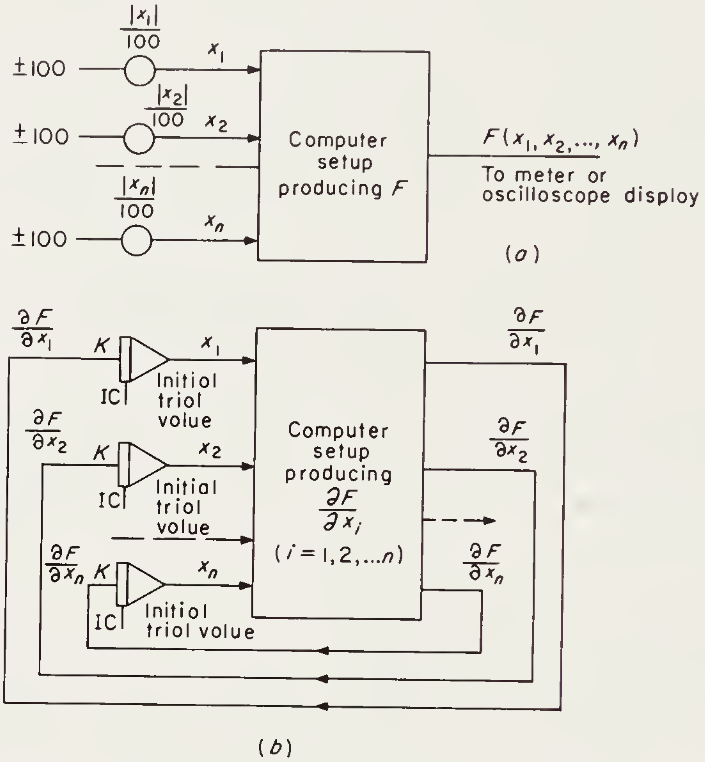

$\frac{|x_1|}{100}$
±100
x_1
$\frac{|x_2|}{100}$
±100
x_2
$\frac{|x_n|}{100}$
±100
x_n
Computer setup producing F
$F(x_1,x_2,\ldots,x_n)$
To meter or oscilloscope display
(a)
$\frac{\partial F}{\partial x_1}$
K
IC
Initial trial value
$\frac{\partial F}{\partial x_2}$
$\frac{\partial F}{\partial x_n}$
x_2
Initial trial value
x_n
Initial trial value
Computer setup producing $\frac{\partial F}{\partial x_i}$ $(i=1,2,\ldots,n)$
(b)

Fig. 8-3. Optimizer with manual parameter adjustments (a) and steepest-descent minimizing scheme (b).

(negative for maximization). Inequality constraints (24) are reduced to ordinary constraints of the form (23), viz.,

$$\left.\begin{aligned} &\varphi_k'(x_1,x_2,\ldots,x_n) \equiv \Psi_k(x_1,x_2,\ldots,x_n)U(\Psi_k) = 0 \\ \text{with}\quad &U(\Psi_k) = \begin{cases} 0 \text{ if } \Psi_k \le 0 \\ 1 \text{ if } \Psi_k > 0 \end{cases} \end{aligned}\right\} \qquad (8\text{-}27)$$

We see that a computer can deal with analytically rather involved questions in a relatively simple-minded way. We also note that any nondecreasing function of F or Φ will be optimized by the same parameter combination $(x_1,x_2,\ldots,x_n)$. This is often convenient; the criterion function (26), for instance, is usually easier to handle with computers, while (25) is simpler analytically.

(b) Automatic Optimization: Continuous Steepest Descent. For continuous automation of our optimization procedure, we replace the parameter-adjusting potentiometers of Fig. 8-3*a* with integrators (Fig. 8-3*b*). Positive or negative integrator inputs will respectively decrease or increase parameter values; for zero input, each integrator will hold the last parameter value. We note that the *i*th integrator input will tend to decrease F if and only if it has the same sign as $\partial F/\partial x_i$; in the two-parameter example of Fig. 8-4, this drives the parameter-space point (x_1,x_2) "downhill." If all n integrators in Fig. 8-3*b* are driven so that

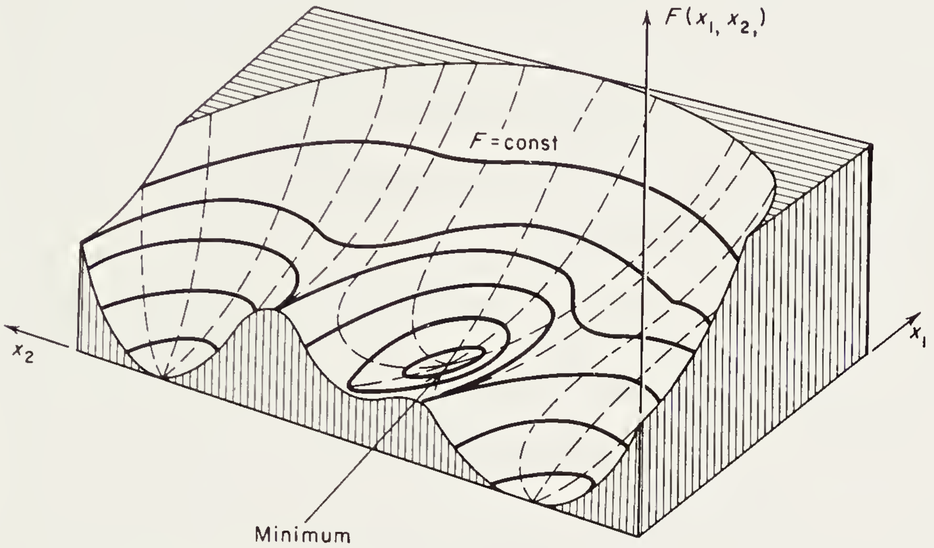


Fig. 8-4. Criterion-function surface for a two-parameter minimum problem, showing three minima, level lines, and gradient lines. In this example, all three minima of $F(x_1,x_2)$ have the same value zero. (*Based on Ref.* 26.)

$$\frac{dx_i}{d\tau} = -K\frac{\partial F}{\partial x_i} \qquad (K > 0;\ i = 1, 2, \ldots, n) \tag{8-28}$$

the parameter-space "point" $(x_1,x_2, \ldots ,x_n)$ will move so as to decrease the criterion function $F(x_1,x_2 \ldots ,x_n)$ along a *line of steepest descent* (*gradient line*) in parameter space, as shown in Fig. 8-4 for the two-dimensional case. $F(x_1,x_2, \ldots ,x_n)$ will thus decrease until we either reach a minimum of F on a boundary of admissible values for the x_i (*boundary minimum*), or until we reach an *interior minimum* with

$$\frac{\partial F}{\partial x_1} = \frac{\partial F}{\partial x_2} = \cdots = \frac{\partial F}{\partial x_n} = 0 \tag{8-29}$$

With multiple minima, each additional minimum must be found by a renewed search starting with a different parameter combination. Note that there is no guarantee of convergence toward the smallest of a set of minima; different minima must be compared after all have been located. Frequently, however, we are looking only for a single minimum, whose location may be approximately known. Maxima of $F(x_1,x_2 \ldots ,x_n)$ are similarly found with $K < 0$, or as minima of $-F$.

Large values of the integrator gain $|K|$ in the optimizing feedback loops (i.e., fast integration) will speed convergence. In practice, computer-system time lags may cause the parameter values to "overshoot" the desired optimum point, so that excessively large values of $|K|$ can cause oscillations. We can prevent this by reducing $|K|$, by replacing the criterion function with one having smaller derivatives near the minimum, or by making K a function of the $\partial F/\partial x_i$ and their time derivatives. In the important special cases where the value of the desired minimum of $F(x_1,x_2, \ldots ,x_n)$ is known to be *zero* (e.g., in equation solvers, Sec. 8-8), we can simply switch all integrators into HOLD when F goes through zero.

Important applications of the steepest-descent technique are described in Secs. 8-8 and 8-9. Another interesting example is the analog computation of eigenvectors $(x_1,x_2, \ldots ,x_n)$ and eigenvalues[130] of a matrix $[a_{ik}]$ through minimization or maximization of the quadratic form $\sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik}x_ix_k$ subject to suitable constraints.[24]

Note that continuous steepest descent requires *explicit generation* of the derivatives $\partial F/\partial x_i$. If these are not known or cannot be readily generated, we must resort to the slower but more generally applicable optimization methods of Secs. 8-11, 8-14, and 8-15. In some applications, optimization setups based on

$$\frac{dx_i}{d\tau} = -K \operatorname{sign} \left(\frac{\partial F}{\partial x_i}\right) \qquad (i = 1,2, \ldots ,n) \tag{8-30}$$

will also be convenient.

8-8. Solution of Equations. (a) Algebraic and Transcendental Equations.

Analog-computer solution of equations

$$\varphi(x) = 0 \tag{8-31}$$

or systems of equations makes sense mainly when we require quick information on the effect of parameter changes on the roots x or when a voltage proportional to a root is required in simulation work. The inverse-function technique of Fig. 4-32 can yield real roots of suitable equations (31); another perfectly feasible solution method[26] sweeps x rapidly through its range and samples x whenever a computer-generated voltage $\varphi(x)$ goes through zero. In general, we prefer to solve Eq. (31) for real roots by minimizing $\frac{1}{2}\varphi^2(x)$ or $|\varphi(x)|$; we implement

$$\frac{dx}{d\tau} = -K\varphi(x)\frac{\partial \varphi}{\partial x} \qquad \text{or} \qquad \frac{dx}{d\tau} = -K\frac{\partial \varphi}{\partial x} \operatorname{sign} \varphi(x) \qquad (K > 0) \tag{8-32}$$

Figure 8-5 illustrates the behavior of $\varphi^2(x)$ near real roots and indicates that our search must start in a suitable neighborhood of each root.

To find complex as well as real roots of an algebraic equation

$$\left.\begin{aligned} &\varphi(z) \equiv a_n z^n + a_{n-1} z^{n-1} + \cdots + a_0 = 0 \\ &\text{with} \\ &z = x + jy = r(\cos\vartheta + j\sin\vartheta) = re^{j\vartheta} \end{aligned}\right\} \tag{8-33}$$

we minimize

$$F \equiv [\operatorname{Re}\varphi(z)]^2 + [\operatorname{Im}\varphi(z)]^2$$

or

$$F \equiv |\operatorname{Re}\varphi(z)| + |\operatorname{Im}\varphi(z)| \tag{8-34}$$

as a function of the real variables x, y or r, ϑ. Further details will be found in Refs. 21 to 28, which also discuss a number of special-purpose equation solvers.

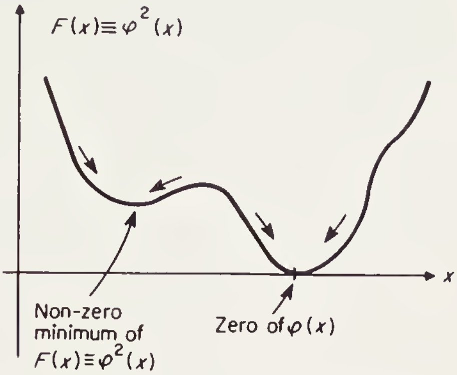


Fig. 8-5. Solution of an equation $\varphi(x) = 0$ by minimization of $F(x) = \varphi^2(x)$, showing a false minimum.

(b) Simultaneous Linear Equations.[21–28] The important problem of solving a set of simultaneous linear equations

$$\sum_{k=1}^{n} a_{ik} x_k + b_i = 0 \qquad (i = 1,2, \ldots ,n) \tag{8-35}$$

is suitable for analog computation if n is not too large; $n = 10$ is reasonable, and $n = 20$ is feasible. We introduce the *residuals* (errors)

$$f_i = \sum_{k=1}^{n} a_{ik} x_k + b_i \qquad (i = 1,2, \ldots ,n) \tag{8-36}$$

Several solution methods exist.

1. *Steepest-descent minimization* of $F(x_1,x_2, \ldots ,x_n) \equiv \frac{1}{2} \sum_{i=1}^{n} f_i^2$ calls for a computer setup for the n differential equations

$$\frac{dx_j}{d\tau} = -K \frac{\partial F}{\partial x_j} \equiv -K \sum_{i=1}^{n} a_{ij} \left(\sum_{k=1}^{n} a_{ik} x_k + b_i \right)$$
$$(K > 0;\ j = 1,2, \ldots ,n) \tag{8-37}$$

   This requires n summing integrators, up to n phase inverters, and up to $2n^2 + n$ coefficient potentiometers (two for each a_{ik} different from 0 or 1).

2. *Manual trial-and-error minimization* or *iterative optimization* of $F(x_1,x_2, \ldots ,x_n) \equiv \frac{1}{2} \sum_{i=1}^{n} f_i^2$ or $F(x_1,x_2, \ldots ,x_n) \equiv \sum_{i=1}^{n} |f_i|$ is often preferable. This requires only up to $n^2 + n$ potentiometers; we

shall need n squarers or absolute-value circuits and an output amplifier to produce F.

These methods converge, in principle, whenever a solution exists, since the normal-mode poles of the linear system (37) correspond to the necessarily nonpositive eigenvalues of the symmetrical matrix $-[a_{ji}][a_{ik}]$.[26] Solution accuracies depend on the nature of the coefficients a_{ik}, with errors below 0.1 to 0.5 percent of half-scale typical for 0.01 percent resistance networks. Some steepest-descent setups may exhibit high-frequency instability.

If the matrix $[a_{ik}]$ is positive-definite,[130] computer setups for

$$\frac{dx_i}{d\tau} = -Kf_i = -K\sum_{k=1}^{n} a_{ik}x_k + b_i \qquad (K > 0,\ i = 1,2,\ \ldots\ ,n) \qquad (8\text{-}38)$$

will produce integrator outputs converging to the desired solution with only $n^2 + n$ potentiometers, again subject to possible high-frequency instability.[28] We can, instead, manually or automatically null f_1 by varying x_1, then null f_2 by varying x_2, etc. (*Gauss-Seidel iteration*).[26] We may be able to improve convergence by reordering and/or rewriting the given equations (35) so as to obtain as large positive diagonal terms a_{ii} as possible.

(c) Hybrid Iteration. Given an analog-computer approximation ${}^1x_k = x_k - e_k$ to the correct solution x_k $(k = 1,2,\ \ldots\ ,n)$ of a suitable linear system (35), then the unknown corrections e_k must satisfy the linear system

$$\left.\begin{array}{lll} \displaystyle\sum_{k=1}^{n} a_{ik}({}^1x_k + e_k) + b_i = 0 & \text{or} & \displaystyle\sum_{k=1}^{n} a_{ik}e_k + {}^1f_i = 0 \\ \text{where} & \displaystyle {}^1f_i = \sum_{k=1}^{n} a_{ik}{}^1x_k + b_i & \end{array}\right\} \quad (i = 1,2,\ \ldots\ ,n) \qquad (8\text{-}39)$$

It may be convenient to compute the residuals ${}^1f_i$ accurately with a digital computer (even a desk calculator). We can then use the analog computer, with unchanged coefficients a_{ik}, to approximate the corrections e_k, using as large scale factors for the ${}^1f_i$ and e_k as possible. We thus obtain a new approximation ${}^2x_k$; we digitally compute new residuals ${}^2f_i = \sum_{k=1}^{n} a_{ik}{}^2x_k + b_i$ and repeat, until all ${}^rf_i$ are as small as required.

8-9. Steepest Descent and Dynamical Constraints.[29–39] **(a) Dynamical Constraints.** A steepest-descent setup can "track" a parameter combination $x_1, x_2, \ldots, x_n$ satisfying a compatible system of equations even when these equations depend on the time t, i.e.,

$$\varphi_k(x_1,x_2,\ \ldots\ ,x_n;t) = 0 \qquad (k = 1,2,\ \ldots\ ,m) \qquad (8\text{-}40)$$

provided that the optimization-loop responses are sufficiently fast. More generally, let $x_1, x_2, \ldots, x_n$ be dynamical variables (state variables) known to satisfy given differential equations

$$\frac{dx_i}{dt} = f_i(t;x_1,x_2,\ \ldots\ ,x_n) + u_i \qquad (i = 1,2,\ \ldots\ ,n) \qquad (8\text{-}41)$$

and let the $u_i = u_i(t)$ be forcing terms to be chosen so as to enforce a compatible set of constraints (40) at all times t, given that the constraints hold for $t = 0$. Assuming that a solution with the desired properties exists, a computer setup implementing

$$u_i = -K\frac{\partial F}{\partial x_i} \qquad (K > 0;\ i = 1,2, \ldots ,n) \tag{8-42}$$

with sufficiently large K and

$$F(x_1,x_2, \ldots ,x_n;t) = \tfrac{1}{2}\sum_{k=1}^{m} c_k \varphi_k^2$$

$$\text{or } F(x_1,x_2, \ldots ,x_n;t) \equiv \sum_{k=1}^{m} c_k \left|\varphi_k\right| \qquad (c_k > 0;\ k = 1,2, \ldots ,m) \tag{8-43}$$

will enforce the given relations (40) between the system variables x_i; *we must adjust the initial values* $x_i(0)$ *so that they satisfy the constraints for* $t = 0$. The stability of such computer setups is discussed in Ref. 39.

In most conventional simulations, constraints are applied directly to eliminate variables, or new constraint-free coordinates are introduced. *The steepest-descent technique, however, is especially useful for adding constraints to dynamical systems already set up on an analog computer.*

In mechanics, fixed constraints

$$\varphi_k(x_1,x_2, \ldots ,x_n) = 0 \qquad (k = 1,2, \ldots ,m) \tag{8-44}$$

to the generalized coordinates x_i of a dynamical system lead to equations of motion of the form

$$\frac{d\dot{x}_i}{dt} = f_i(t;x_1,x_2, \ldots ,x_n;\dot{x}_1,\dot{x}_2, \ldots ,\dot{x}_n) + u_i \qquad \frac{dx_i}{dt} = \dot{x}_i \qquad (i = 1,2, \ldots ,n) \tag{8-45}$$

where the $u_i = u_i(t)$ are again forcing terms required to enforce the given constraints (44). Since forces due to fixed constraints cannot do work, they must be perpendicular to all displacements permitted by the constraints; it follows that

$$u_i = \sum_{k=1}^{m} \lambda_k(t)\frac{\partial \varphi_k}{\partial x_i} \qquad (i = 1,2, \ldots ,n) \tag{8-46}$$

where the $\lambda_k(t)$ are uniquely determined by Eqs. (44) and (45).[130] Each term in Eq. (46) can be interpreted in terms of physical generalized-force components due to the constraints.

In analog-computer simulation, the $\lambda_k(t)$ are rarely known explicitly. We approximate the correct expressions (46) by heuristic steepest-descent methods, which often admit physical interpretation. We may, for instance, attempt to minimize

$$F(x_1,x_2, \ldots ,x_n) = \sum_{k=1}^{m} c_k\varphi_k^2 \qquad (c_k > 0,\ k = 1,2, \ldots ,m) \tag{8-47}$$

by taking

$$u_i = -K\frac{\partial F}{\partial x_i} = -2K\sum_{k=1}^{m} c_k\varphi_k\frac{\partial \varphi_k}{\partial x_i} \qquad (K > 0;\ i = 1,2, \ldots ,n) \tag{8-48}$$

which can be interpreted as a sort of elastic reaction resisting any violation of our constraints. We can, instead, try to minimize

$$F(x_1,x_2,\ldots,x_n;\dot{x}_1,\dot{x}_2,\ldots,\dot{x}_n) = \sum_{k=1}^{m} c'_k\dot{\varphi}_k^2 \qquad (c'_k > 0,\ k = 1,2,\ldots,m)$$

by

$$u_i = -K'\frac{\partial F}{\partial \dot{x}_i} = -2K'\sum_{k=1}^{m} c_k\dot{\varphi}_k\frac{\partial\dot{\varphi}_k}{\partial \dot{x}_i} \qquad (K' > 0;\ i = 1,2,\ldots,n) \tag{8-49}$$

which corresponds roughly to viscous reactions against constraint violation and has less tendency to "elastic" oscillations. We can also combine constraint "forces" of the forms (48) and (49). The reader should note carefully that such steepest-descent methods are *necessarily approximate*. The correct forcing terms (46) do *not* in general vanish when all constraints are satisfied, while our approximations (48) and (49) require small violations of the constraints to generate the desired forcing terms. In a sense, this is a realistic simulation of forces resisting deformations of actual constraining rods, strings, tracks, etc. Assuming true *equality* constraints satisfied initially by all x_i, and a stable computer setup with large K or K', the exact form of F will not matter too much. The case of *inequality constraints* (e.g., limit stops, barriers, loose strings) is more complicated, and F must be carefully chosen to fit the energy considerations of the problem. As in the case of limit stops (Sec. 8-24), the method of Fig. 8-31 may yield more accurate results.

(b) Steepest-descent Methods for Improving Differential-analyzer Accuracy.$^{34-40}$ Steepest-descent implementation of suitable constraints often produces *quotients* and *inverse functions* more accurately and stably than the high-gain-amplifier circuits of Figs. 4-32 and 6-23, although slightly more equipment is required. In particular, the *four-quadrant division circuit* of Fig. 8-6*a* produces $Z = 100X/Y$ by continuously minimizing

$$F(X,Y,Z) \equiv \frac{1}{2}\,\varphi^2(X,Y,Z) = \frac{1}{2}\left(\frac{YZ}{100} - X\right)^2 \tag{8-50}$$

The circuit implements

$$\frac{dZ}{d\tau} = -K\frac{\partial F}{\partial Z} \equiv -K\frac{Y}{100}\left(\frac{YZ}{100} - X\right) \tag{8-51}$$

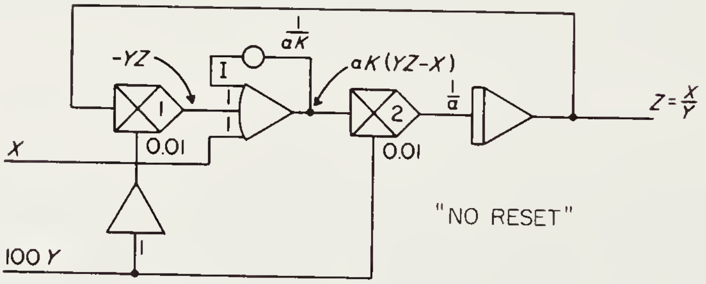


Fig. 8-6*a*. This four-quadrant division circuit produces $Z = X/Y$ by continuously minimizing $F = \frac{1}{2}(YZ - X)^2$. For constant Y, Z follows X with a lag time constant equal to the reciprocal loop gain $1/KY^2$, which is ordinarily quite small. The circuit permits computation of $\lim_{\tau\to t_1}[X(\tau)/Y(\tau)]$ with $X(\tau_1) = Y(\tau_1) = 0$, if the limit exists. Note that the initial output $Z(0) = X(0)/Y(0)$ is correctly produced by the steepest-descent integrator, which is always in the COMPUTE mode.

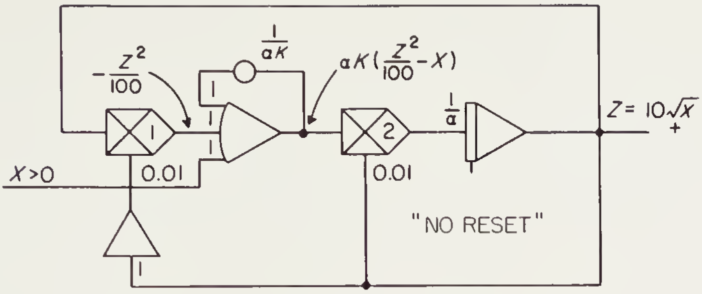


Fig. 8-6*b*. Steepest-descent square-root circuit.

and works with divisors Y of either polarity; X and Y can even go through zero together. Figure 8-6*b* shows a steepest-descent *square-root circuit*, which continuously minimizes $\frac{1}{2}\varphi^2(X,Z) = \frac{1}{2}(Z^2/100 - X)^2$. *Note that the steepest-descent integrators in Fig.* 8-6*a and b are never reset and automatically supply the correct initial conditions when the rest of the computer is in the* RESET *mode.* Moreover, no precision feedback capacitors are needed, so that high-gain amplifiers with plug-in capacitors can serve as integrators. Capacitances will be as small as stability allows.

The extra analog multipliers required for the steepest-descent circuits in Fig. 8-6*a* and *b* can be replaced by simple analog switches if we minimize the absolute value $|\varphi|$ instead of $\frac{1}{2}\varphi^2$. The corresponding machine equation for steepest-descent division is

$$\frac{dZ}{d\tau} = -K\frac{Y}{100}\operatorname{sign}\left(\frac{YZ}{100} - X\right) \tag{8-52a}$$

Steepest-descent division circuits implementing Eq. (51) or (52*a*) are feedback loops designed to follow the input X with a loop gain proportional to the divisor Y. It follows that bandwidth and transient response with respect to X deteriorate as $|Y|$ decreases. This situation is improved by steepest-descent division circuits based on[37,40]

$$\frac{dZ}{d\tau} = -K\left(\frac{YZ}{100} - X\right)\operatorname{sign} Y \tag{8-52b}$$

or even on

$$\frac{dZ}{d\tau} = -K\operatorname{sign}\left(\frac{YZ}{100} - X\right)\operatorname{sign} Y \tag{8-52c}$$

which produce absolute loop gains independent of Y. Multiplication by a sign function, less expensive than true analog multiplication, can be implemented by simple comparator/logic-driven switches or bridge circuits (Sec. 4-8 and Fig. 8-7). The resulting steepest-descent loops have a tendency toward small limit-cycle oscillations; a small comparator deadspace may be helpful.

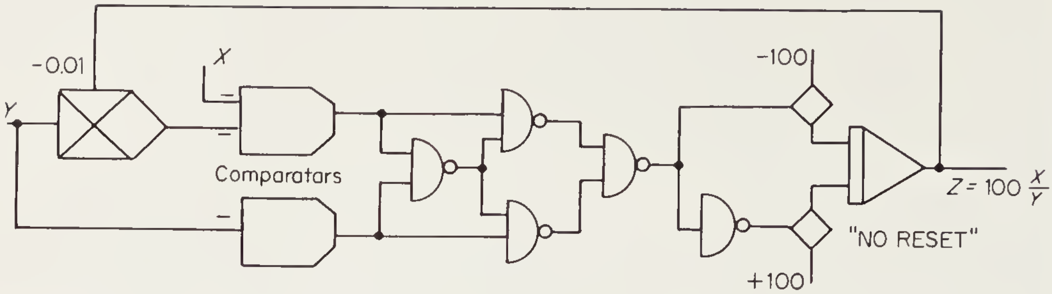


Fig. 8-7. A steepest-descent division circuit using comparators and logic to implement Eq. (52*c*).

Another potentially highly useful application of the steepest-descent technique is the *enforcement of known properties of a computer solution as redundant constraints implemented just as in Sec. 8-9a.* The best-known application is the amplitude stabilization of rate resolvers and sine-wave oscillators by continuous minimization of $(X^2 + Y^2 - R^2)^2$, where X and Y are differential-analyzer solutions $X = R\cos\vartheta$, $Y = R\sin\vartheta$ (Sec. 6-24). Exact computer solutions X, Y would satisfy the constraint in any case (redundant constraint), but the steepest-descent

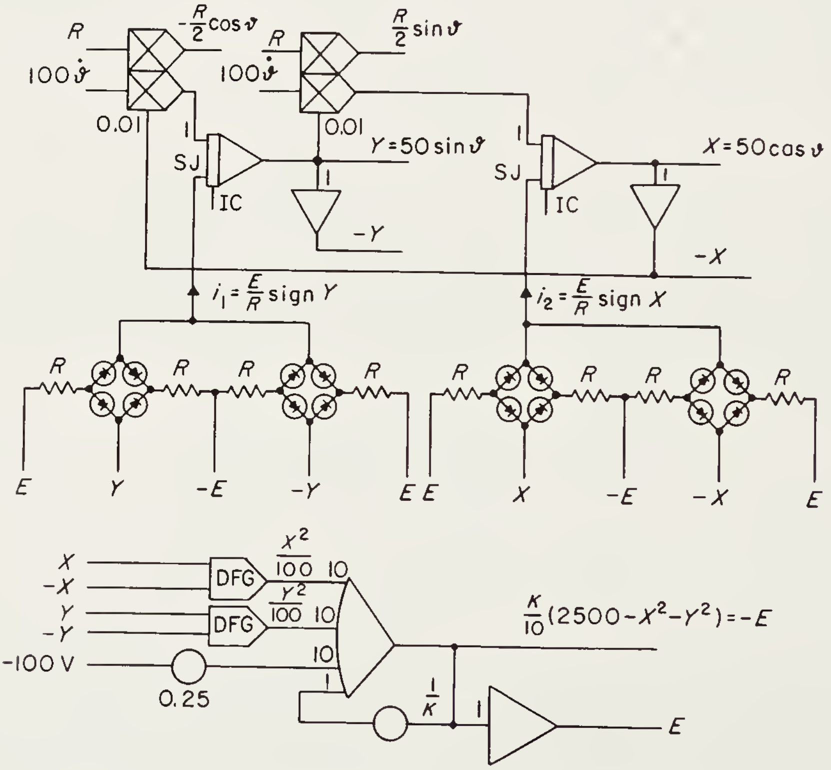


Fig. 8-8. Electronic resolver with amplitude correction (see also Sec. 6-24). The feedback currents EX, EY are monotonic functions of EX, EY, so that the circuit continuously minimizes $(X^2 + Y^2 - 2{,}500)^2$.

feedback continuously corrects the effects of small phase-shift errors and thus improves the solution accuracy very significantly.

The main objection to the resolver circuit of Fig. 6-34*b* is the relatively large number of multipliers required. The simplified stabilized resolver of Fig. 8-8 replaces the two multipliers required to generate EX, EY by sign multipliers producing E sign X, E sign Y. Since these simplified corrections are monotonic functions of EX, EY, they will minimize $(X^2 + Y^2 - 2{,}500)^2$. Since EX, EY do not switch abruptly when E goes through zero, there is less tendency toward limit-cycle oscillations than in earlier circuits employing sign-multiplier outputs X sign E, Y sign E to minimize $|X^2 + Y^2 - 2{,}500|$.[32,37]

8-10. Linear Programming and Related Problems.[23–26] *Linear-programming problems* in operations analysis, game theory, and economics require the determination of *a set of* n *numbers* $x_1, x_2, \ldots, x_n$ *which minimizes (or maximizes) a linear expression*

$$v(x_1,x_2, \ldots ,x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{8-53}$$

while satisfying $m \geq n$ *linear inequalities*

$$y_i \equiv a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - y_{i0} \geq 0 \qquad (i = 1,2, \ldots ,m) \tag{8-54}$$

In a typical application, the problem is to buy necessarily positive quantities x_k of n types of raw materials ("inputs") so as to minimize the cost (53) while keeping the respective quantities y_i of $m - n$ output products at or above $m - n$ specified levels y_{i0} $(i = n + 1, n + 2, \ldots, m)$. Note that, since $x_i \geq 0$ $(i = 1,2, \ldots ,n)$, we actually have m inequalities of the form (54).

The linear-programming problem defined by Eqs. (53) and (54) is equivalent to the minimization (or maximization) of the function

$$\left.\begin{aligned} &F(x_1,x_2, \ldots ,x_n) \equiv v + \lambda_1 |y_1| + \lambda_2 |y_2| + \cdots + \lambda_m |y_m| \\ \text{with} \qquad &\lambda_i = \begin{cases} 0 \ (y_i \geq 0) \\ B \ (y_i < 0) \end{cases} \qquad (i = 1,2, \ldots ,m) \end{aligned}\right\} \tag{8-55}$$

where B is a large positive constant. Assuming the existence of a solution, the analog computer minimizes $F(x_1,x_2, \ldots ,x_n)$ by implementing the n differential equations

$$\frac{dx_k}{d\tau} = -\alpha \frac{\partial F}{\partial x_k} = -\alpha(c_k + a_{1k}\lambda_1 + a_{2k}\lambda_2 + \cdots + a_{mk}\lambda_m) \qquad (k = 1,2, \ldots ,n) \tag{8-56}$$

where α is a positive constant, and τ is the computer time (real time). The resulting steepest-descent feedback will decrease $F(x_1,x_2, \ldots ,x_n)$ and hence minimize v subject to the given constraints (54).

In the example of Fig. 8-9*a*, x_1 and x_2 may be regarded as the rectangular cartesian coordinates of a particle moving in a viscous medium. The constraints are satisfied for all points (x_1,x_2) above and to the right of the convex polyhedron of smooth straight-line barriers indicated in heavy lines. The broken lines are lines of equal potential energy $v(x_1,x_2)$. The particle will move in the negative-gradient direction until it strikes a barrier, say by violating the inequality $y_4 \geq 0$. The particle then experiences an additional force perpendicular to the barrier. As a result, the particle will move along the heavy lines to the barrier vertex with the lowest potential energy $v(x_1,x_2)$. The problem is degenerate if ∇v is perpendicular to a barrier; in this case, the solution may not be unique.

Maximization problems would be solved similarly, with the minus sign in Eq. (56) replaced by a plus sign.

Figure 8-9*b* shows the analog-computer setup for the steepest-descent equations (56). The problem variables x_k, y_i, and v are represented by corresponding voltages $X_k = a_k x_k$, $Y_i = b_i y_i$, so that Eqs. (54) and (56)

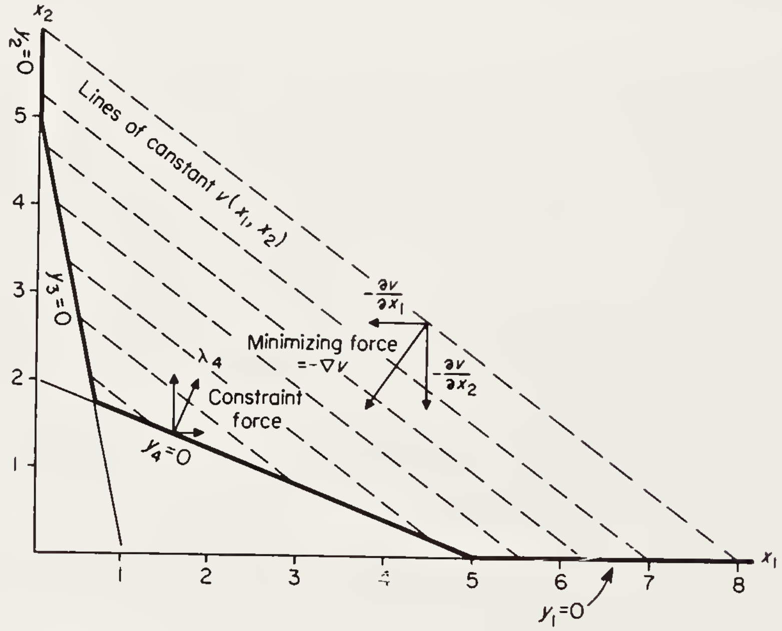

x_2
$y_2=0$
5
4
3
2
1
Lines of constant $v(x_1, x_2)$
$-\frac{\partial v}{\partial x_1}$
Minimizing force $=-\nabla v$
$-\frac{\partial v}{\partial x_2}$
λ_4
Constraint force
$y_3=0$
$y_4=0$
1
2
3
4
5
6
7
8
x_1
$y_1=0$

Fig. 8-9*a*. Dynamical-analogy interpretation of a linear-programming problem: steepest-descent minimization of

$$v(x_1,x_2) = \tfrac{1}{4}x_1 + \tfrac{1}{3}x_2$$

subject to the constraints

$$y_1 \equiv x_1 \geq 0 \qquad y_2 \equiv x_2 \geq 0$$

$$y_3 \equiv 10x_1 + 2x_2 - 10 \geq 0 \qquad y_4 \equiv 2x_1 + 5x_2 - 10 \geq 0$$

Constraint "forces" prevent penetration of "barriers" corresponding to each constraint inequality.

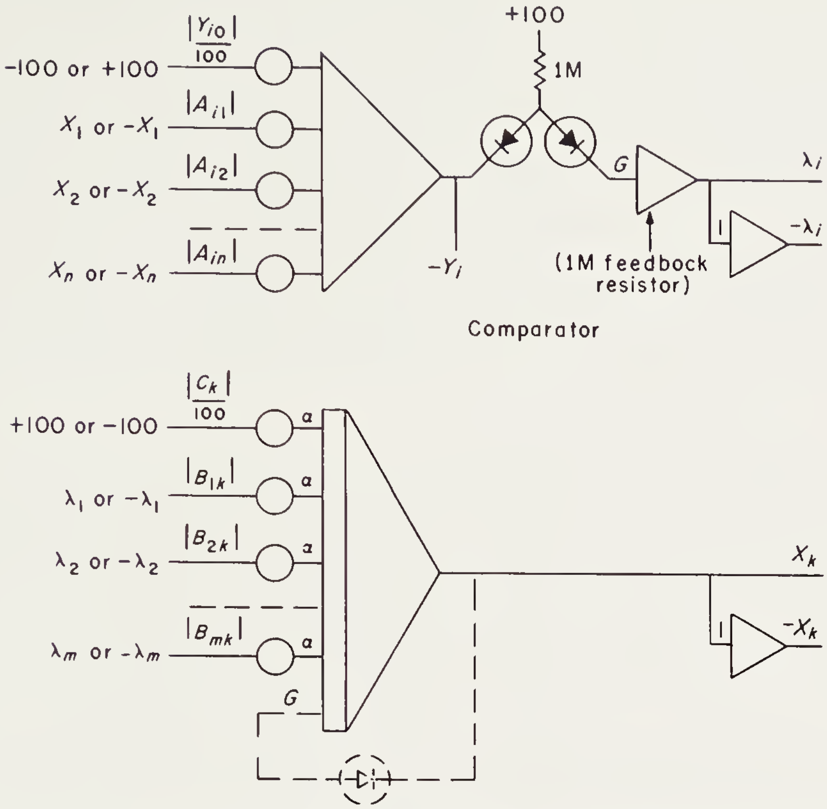


Fig. 8-9*b*. Analog-computer setup for the solution of linear-programming problems. The inverted input voltages $-X_k$, $-\lambda_i$, or -1 are used with negative coefficients A_{ik} or C_k. If, as in many practical problems, an inequality $Y_i \geq 0$ takes the simple form $X_j \geq 0$, then the entire block of computing elements generating the corresponding voltage λ_i is simply replaced by a feedback diode connected across the *j*th integrator so as to limit its output to nonnegative values (dashed lines).

take the form of machine equations

$$\left.\begin{aligned} Y_i &= A_{i1}X_1 + A_{i2}X_2 + \cdots + A_{in}X_n - Y_{i0} \\ PX_k &= -\alpha(C_k + B_{1k}\lambda_1 + B_{2k}\lambda_2 + \cdots + B_{mk}\lambda_m) \\ \text{with} \qquad C_k &= a_k c_k \qquad A_{ik} = \frac{b_i}{a_k} a_{ik} \qquad B_{ik} = a_k a_{ik} \\ Y_{i0} &= b_i y_{i0} \qquad (i = 1,2, \ldots ,m;\ k = 1,2, \ldots ,n) \end{aligned}\right\} \tag{8-57}$$

m voltages $-Y_i$ are produced by summing amplifiers and applied to analog comparators generating the constraint-free voltages λ_i. Each X_k is obtained at the output of a summing integrator. Phase inverters also produce the voltages $-\lambda_i$ and $-X_k$ as needed for multiplication by negative coefficients. Coefficients exceeding unity in absolute value are obtained with the aid of amplifier input gains in the usual manner. Solutions converge within fractions of a second, depending on the integrator gain α used. A voltage proportional to v, if required, could be readily generated with an additional summing amplifier.

The analog-computer method is practical for linear-programming problems with $m, n \leq 20$; the practically instantaneous solution makes it particularly convenient to study effects of coefficient changes on the values of v and of the "slack variables" y_i. The n variables y_i closest to zero correspond to those of the m inequalities (54) which actually determine the solution vertex (see also Fig. 8-9*a*). One can, then, obtain solutions of essentially unlimited accuracy by solving the corresponding n simultaneous *equations* $y_i = 0$ by the hybrid iteration method of Sec. 8-8*c*. Our analog-computer technique is easily extended to suitable, more general programming involving *nonlinear* objective functions and constraints.[25]

8-11. Optimization by Parameter Perturbation. Although direct implementation of the steepest-descent relations (28) is successful in suitable applications, one must remember that computer generation of the required derivatives $\partial F/\partial x_k$ may well be impractically complicated or, indeed, impossible. In particular, the function $F(x_1,x_2, \ldots ,x_n)$ might be known only as the result of measurements or partial system tests, rather than as an explicit differentiable relationship. In such situations, we may replace the derivatives $\partial F/\partial x_k$ with measured ratios $\delta F/\delta x_k$ corresponding to small changes δx_k produced either by normal system operation or by test-signal injection (*parameter perturbation*).

Referring to Fig. 8-10, a sinusoidal perturbation $\delta x_1 = a \cos \omega\tau$ causes a sinusoidal variation

$$\delta F \approx \frac{\partial F}{\partial x_1}\,\delta x_1 = \frac{\partial F}{\partial x_1}\,a \cos \omega\tau$$

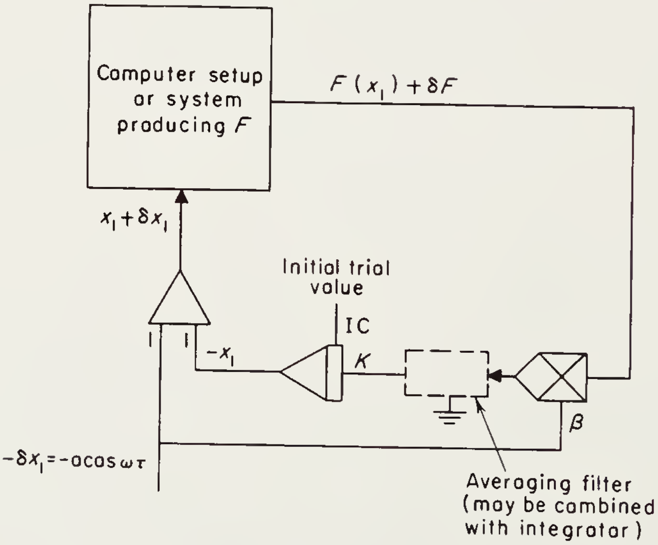

Computer setup or system producing F
$F(x_1)+\delta F$
$x_1+\delta x_1$
1
1
$-x_1$
Initial trial value
IC
K
β
$-\delta x_1=-a\cos\omega\tau$
Averaging filter (may be combined with integrator)

Fig. 8-10. A simple parameter-perturbation optimizer.

This is multiplied by δx_1 to produce

$$\delta F\, \delta x_1 \approx \frac{\partial F}{\partial x_1} a^2 \cos^2 \omega\tau = \frac{a^2}{2} \frac{\partial F}{\partial x_1} + \frac{a^2}{2} \frac{\partial F}{\partial x_1} \cos 2\omega\tau$$

which is averaged and inverted to yield the desired steepest-descent feedback proportional to $-(\partial F/\partial x_1)$. It is convenient to replace the sinusoidal perturbation by a square-wave signal, so that the multiplication reduces to simple switching. Depending on the nature of our system, it is usually possible to optimize two or more parameters x_k simultaneously with noninterfering (orthogonal) perturbation signals (typically square waves or sine waves of different frequencies), at the price of slower convergence.[15,16]

If $F(x_1,x_2, \ldots ,x_n)$ is a dynamical-system output rather than just a "static" function of the x_k, then the periodic variation δF will lag the periodic input δx_1. For this reason, Fig. 8-10 includes a phase-shifting circuit (not shown) to be adjusted for optimum feedback. From a more sophisticated point of view, our parameter-perturbation-optimizer feedback approximates a value of the time crosscorrelation function of δx_1 and F, which must vanish for $\partial F/\partial x_1 = 0$.[82] The artificial parameter perturbation is felt as noise in the system output, and the normal system input similarly causes interference in the optimizer loop. The choice of the perturbation frequency involves a compromise between optimizer-loop response and minimum interference.[42] It is also possible to replace periodic parameter perturbation with a noiselike signal.

8-12. Continuous Optimization and Model Matching. Our technique for enforcing dynamical constraints (40) on time-varying dynamical variables $x_1, x_2, \ldots , x_n$ by steepest descent was interpreted as continuous minimization of a criterion function (error measure) (43). The question naturally arises whether similar steepest-descent techniques can continuously minimize more general criterion functions. In actual fact, such problems are only rarely meaningful. In the case of our compatible constraints, an actual minimum corresponding to $F = 0$ *can* be realized at all times, since we were careful to assume all constraints to be satisfied at $t = 0$. In general, the situation is not so simple; the best we can usually do is to find control functions $u_i(t)$ which will minimize a time average, weighted time average, or running average of F. This, however, is a problem in *functional* optimization (Secs. 8-13 and 8-16); the technique of Sec. 8-9*a* may or may not yield a useful approximation.

A very important application *apparently* calling for continuous minimization is the problem of *model matching*. Referring to Fig. 8-11, we should like to match the parameters $\alpha_1, \alpha_2, \ldots , \alpha_n$ of a computer-represented mathematical model to its prototype (real system or computer setup) by continuously minimizing an error measure, say,

$$F(\alpha_1,\alpha_2, \ldots ,\alpha_n) = \tfrac{1}{2} \sum_{k=1}^{m} c_k[y_k(t;\alpha_1,\alpha_2, \ldots ,\alpha_n) - \eta_k(t)]^2 \qquad (c_k \geq 0;\ k = 1,2, \ldots ,m) \tag{8-58a}$$

as both model and prototype are subjected to the same test input or inputs. The latter must be chosen with some care so as to make the error reasonably sensitive to variations in each unknown parameter.[42,86] Since not all the state variables η_k may be observable, some of the c_k in Eq. (58*a*) may have to be zero, which can make identification of certain parameters difficult or impossible.

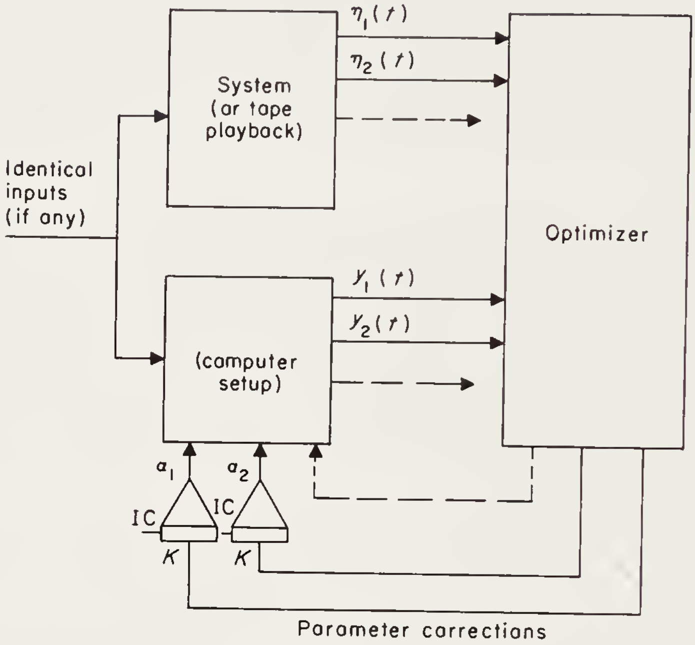


Fig. 8-11. Model matching.

With a model described by m differential equations

$$\frac{dy_i}{dt} = f_i(t; y_1, y_2, \ldots, y_m; \alpha_1, \alpha_2, \ldots, \alpha_n) \qquad (i = 1,2, \ldots, m) \tag{8-59}$$

a possibly more useful error measure may be an *equation-error* measure[85] like

$$F(\alpha_1, \alpha_2, \ldots, \alpha_n) = \tfrac{1}{2} \sum_{i=1}^{m} c_i \left[\frac{dy_i}{dt} - f_i(t; y_1, y_2, \ldots, y_m; \alpha_1, \alpha_2, \ldots, \alpha_n) \right]^2$$
$$(c_i \geq 0;\ i = 1,2, \ldots, m) \quad (8\text{-}58b)$$

This method is especially suitable for time-invariant linear systems.[85] Model matching is of the greatest practical importance:

1. For deriving *simplified models* of complicated systems, including biological systems and human operators[42]
2. For *system identification* (*parameter estimation*), especially for adaptive control of systems with time-varying parameters

Conversely, certain adaptive control systems attempt to match parameters of an actual physical system to those of a model with desirable properties.[84]

If the unknown parameters α_k vary with time, continuous minimization of (58) makes sense only in rare cases, as in parameter *tracking*, where system and model satisfy similar differential equations (59) with initially identical parameters.

Purely formally, steepest-descent minimization of (58*a*) would require

$$\frac{d\alpha_k}{dt} = -K\frac{\partial F}{\partial \alpha_k} = -K\sum_{i=1}^{m} c_i(y_i - \eta_i)\frac{\partial y_i}{\partial \alpha_k} \qquad (K > 0;\ k = 1,2, \ldots ,n) \tag{8-60}$$

Several investigators[42,86] have implemented these relations with parameter-influence coefficients $\partial y_i/\partial \alpha_k$ obtained through continuous solution of sensitivity equations

$$\frac{d}{dt}\left(\frac{\partial y_i}{\partial \alpha_k}\right) = \sum_{j=1}^{m}\frac{\partial f_i}{\partial y_i}\frac{\partial y_j}{\partial \alpha_k} + \frac{\partial f_i}{\partial \alpha_k} \qquad (i = 1,2, \ldots ,m) \tag{8-61}$$

in the manner of Sec. 8-4. The varying success[42] of these methods depends on the extent to which they approximate a really meaningful functional minimization; parameter-influence coefficients, in fact, were originally defined for *constant* parameters α_k only.[42,85]

Model matching through *parameter perturbation* (Sec. 8-11) is more easily made consistent with valid theory, for the computer setup of Fig. 8-10 approximates true functional optimization as soon as we insert any kind of short-term averaging filter at the output of the function generator producing the criterion function F. The success of parameter-perturbation model matching, although necessarily subject to compromises between averaging time and accuracy, is thus more easily predictable.[16,81–83]

ITERATIVE OPTIMIZATION

8-13. Parameter Optimization of Functionals: Examples. We turn next to the optimization of *functionals* or numerical functions of the entire time history of a function $y(t)$ (or of several such functions) obtained in the course of a computer run. True functional optimization requires us to select an optimal *function* $y(t)$ (Sec. 8-16), but we start with the important problem of *parameter optimization.* $y(t) = y(t;\alpha_1,\alpha_2, \ldots ,\alpha_n)$ is to be completely determined by n constant parameters $\alpha_1, \alpha_2, \ldots , \alpha_n$, and we want to determine $\alpha_1, \alpha_2, \ldots , \alpha_n$ so as to maximize or minimize a functional dependent on $y(t)$. Simple examples of functionals are $\int_0^T \Phi[y(t)]\,dt$, where $\Phi(y)$ is a given function, or simply $\Phi[y(T)]$; note that *each such functional is simply a numerical function* $F(\alpha_1,\alpha_2, \ldots ,\alpha_n)$ *of the parameters* $\alpha_1, \alpha_2, \ldots , \alpha_n$. In most applications,

$$y(t) = y(t;\alpha_1,\alpha_2, \ldots ,\alpha_n)$$

is not explicitly given as a function of the parameters, but as the solution of a differential equation containing the parameters,

$$\frac{d^r y}{dt^r} = f(t;y,\dot{y},\ddot{y}, \ldots ;\alpha_1,\alpha_2, \ldots ,\alpha_n) \tag{8-62}$$

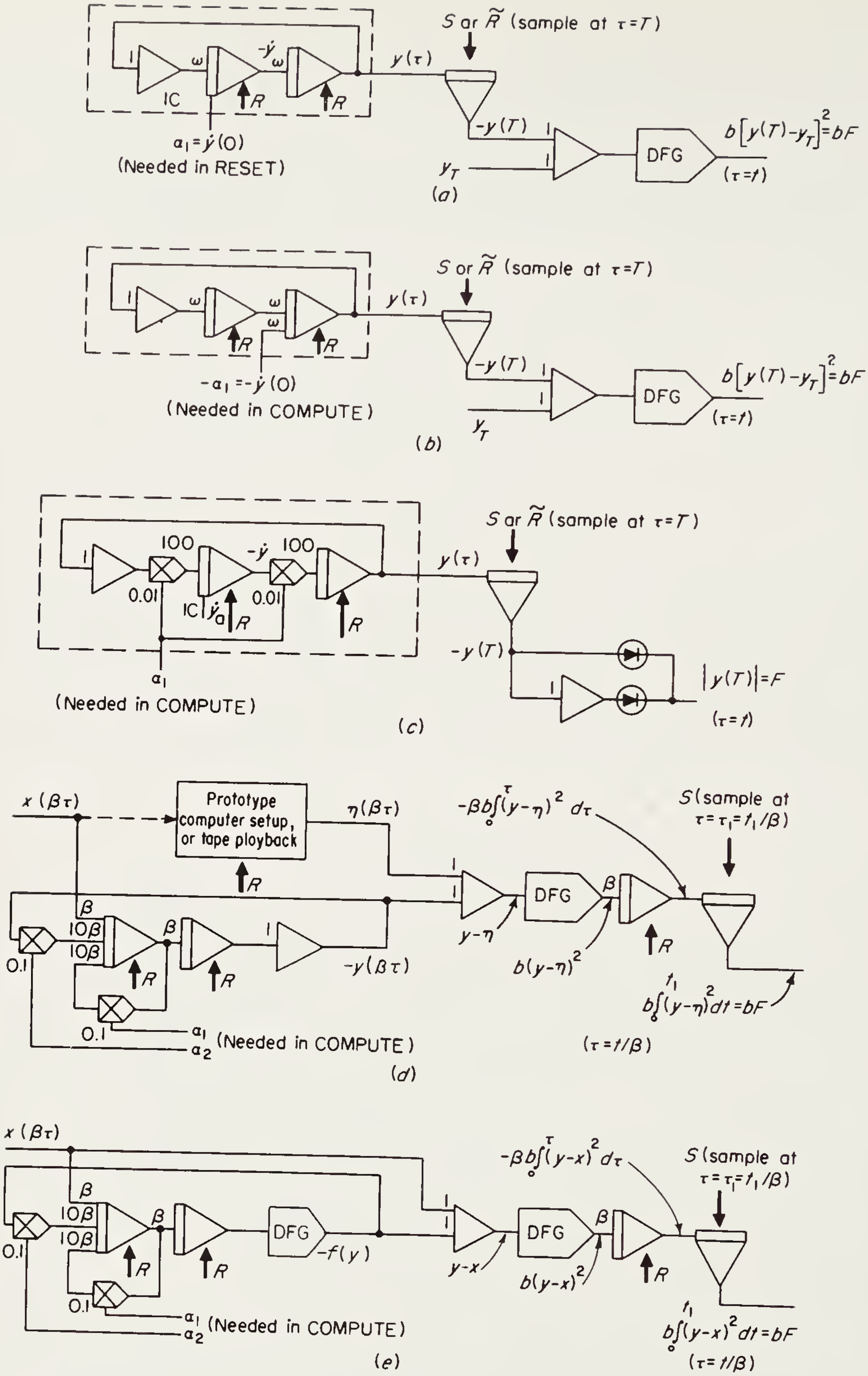


Fig. 8-12. Parameter optimization of functionals. Four typical problems are illustrated: a simple boundary-value problem (*a*, *b*), an eigenvalue problem (*c*), a constant-parameter model-matching problem (*d*), and two-parameter control-system optimization (*e*). If the parameters α_1, α_2 to be optimized are supplied by a digital computer, the analog multipliers shown in (*c*), (*d*), and (*e*) are simply replaced with MDACs.

The unknowns $\alpha_1, \alpha_2, \ldots, \alpha_n$ are, typically, design parameters for a dynamical system described by Eq. (62), and our functional is a performance measure; unknown initial values $y(0), \dot{y}(0), \ldots$, too, can play the role of unknown parameters. The following examples, simple as they are, illustrate the most significant applications:

1. *A Boundary-value Problem* (Fig. 8-12*a*, *b*). The two-point boundary-value problem

$$\frac{d^2y}{dt^2} = -\omega^2 y \qquad y(0) = 0 \qquad y(T) = y_T \qquad \left(T \neq \frac{k\pi}{\omega}, k = 0,1,2, \ldots\right) \tag{8-63}$$

   is conveniently solved by minimization of the functional (error measure)

$$[y(T) - y_T]^2 = F[\dot{y}(0)] \tag{8-64}$$

   as a function of the unknown parameter $\alpha_1 = \dot{y}(0)$.
2. *An Eigenvalue Problem* (Fig. 8-12*c*). Eigenfunctions $y(t)$ and eigenvalues[130] α_1^2 satisfying

$$\frac{d^2y}{dt^2} = -\alpha_1^2 y \qquad y(0) = 0 \qquad \dot{y}(0) = \dot{y}_0 \qquad y(T) = 0 \tag{8-65}$$

   are found by minimization of the functional

$$|y(T)| = F(\alpha_1) \tag{8-66}$$

   as a function of the unknown parameter α_1.
3. *Constant-parameter Model Matching* (Fig. 8-12*d*; see also Sec. 8-12). Two unknown constant parameters α_1, α_2 of a mathematical model defined by

$$\frac{d^2y}{dt^2} + \alpha_1 \frac{dy}{dt} + \alpha_2 y = x(t) \qquad y(0) = \dot{y}(0) = 0 \tag{8-67}$$

   are matched to a physical prototype system (black box) with input $x(t)$ and output $\eta(t)$ through minimization of the error measure

$$\int_0^{t_1} [y(t) - \eta(t)]^2 \, dt = F(\alpha_1, \alpha_2) \tag{8-68}$$

4. *Control-system Optimization* (Fig. 8-12*e*). Optimal constant parameters α_1, α_2 for a simple servomechanism described by

$$\frac{d^2y}{dt^2} + \alpha_1 \frac{dy}{dt} + \alpha_2 f(y) = x(t) \qquad y(0) = \dot{y}(0) = 0 \tag{8-69}$$

for a given input $x(t)$ are determined through minimization of an error measure such as

$$\int_0^{t_1} [y(t) - x(t)]^2\, dt = F(\alpha_1,\alpha_2) \tag{8-70}$$

As a natural generalization, the functional to be optimized can depend on m functions $y_i(t)$ given by m first-order differential equations like Eq. (59), with n unknown constant parameters $\alpha_1, \alpha_2, \ldots, \alpha_n$.

8-14. Iterative-differential-analyzer Techniques. (a) Introduction. Many parameter-optimization problems are conveniently solved with iterative differential analyzers (Sec. 2-2). In each example of Fig. 8-12 successive differential-analyzer runs with new parameter values $\alpha_1\ \alpha_2, \ldots, \alpha_n$ will produce new solutions $y(t)$, and hence new values of the given criterion functional $F(\alpha_1,\alpha_2, \ldots ,\alpha_n)$; the machine acts as a function generator for $F(\alpha_1,\alpha_2, \ldots ,\alpha_n)$, with sampled-data output once per computer run. With sufficiently fast iteration (5 to 1,000 computer runs/sec), a track-hold output representing $F(\alpha_1,\alpha_2, \ldots ,\alpha_n)$ can be displayed on an oscilloscope or meter and will appear to change more or less continuously with parameter settings. Such a display permits *optimization by successive manual parameter adjustments.* With a favorable ("convex") criterion function $F(\alpha_1,\alpha_2, \ldots ,\alpha_n)$ and a fast computer, such trial-and-error iteration can converge rapidly and is simple, practical, and useful. *Automatic iterative optimization* replaces the human operator with suitable optimization logic, i.e., with a sampled-data control system designed to set optimal parameter values. Automatic optimization is needed especially:

1. For rapid plotting of optimal values of F and/or $\alpha_1, \alpha_2, \ldots, \alpha_n$ against other system parameters
2. If the optimal values of F and/or $\alpha_1, \alpha_2, \ldots, \alpha_n$ are to be used as inputs to other computing elements, or to actual physical systems[84]
3. If we want to use a (possibly more accurate) differential analyzer too slow for repetitive-display optimization

Our assumption of a "meaningful" criterion function presupposes the *existence* of an optimal parameter combination, but we may have to deal with more than a single maximum or minimum. As in our example of Fig. 8-4, an iteration scheme attempting to decrease F, say, might lead us to a different minimum depending on our initial trial values for $\alpha_1, \alpha_2, \ldots, \alpha_n$ ("starting point" in parameter space). Our simple eigenvalue problem defined by Eq. (65) is a case in point. Unless all but one of the "optimal" parameter combinations can be eliminated by other considerations, we may have to find two or more maxima or minima and select the best one. With one, two, or even three or four unknown parameters, we may get an impression of the nature of $F(\alpha_1, \alpha_2, \ldots)$ through preliminary computations over a coarse net of parameter values. For a larger number of parameters, however, some analytical or physical a priori knowledge about the location of the desired optimum is virtually indispensable. It is also usually a good idea to recheck each maximum or minimum located through iteration by renewed iteration with a different starting point.

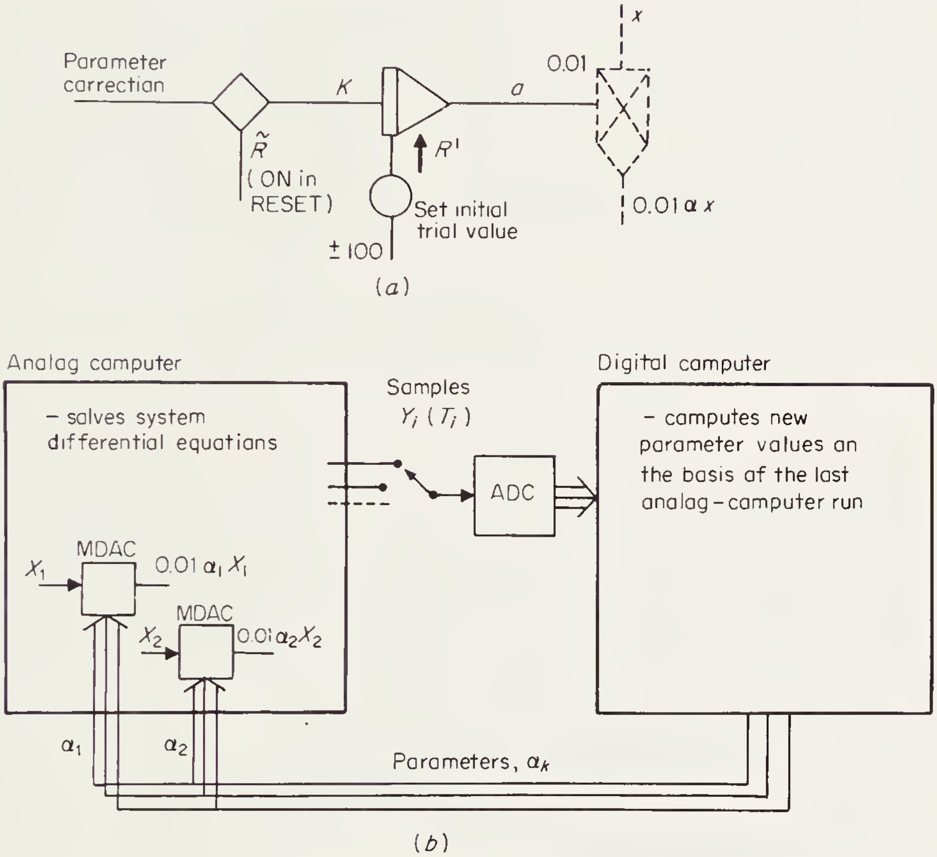


Fig. 8-13. Parameter setting for optimization. The MDACs in (*b*) are updated just before the end of each analog-computer RESET period.

(b) Automatic Iteration. To automate our sampled-data optimization, we can, as in Sec. 8-7, replace each manual parameter adjustment with an integrator (Fig. 8-13*a*). For functional optimization, we place these integrators into HOLD during differential-analyzer COMPUTE periods; the parameter-setting integrators integrate only during computer RESET periods.

This parameter-updating method, with analog-plus-patched-logic computation of parameter corrections, is practical only for very simple optimization studies. *The preferred technique for iterative optimization employs a digital computer, which computes new parameter values and* *sets them into multiplying digital-to-analog converters (MDACs) before the end of each analog-computer* RESET *period*, as in Fig. 8-13*b*. The analog computer serves only for rapid solution of the system differential equations. This is a particularly favorable combination of analog and digital techniques (see also Secs. 2-2 and 9-13).

We denote parameter values during the first, second, . . . computer run by ${}^0\alpha_k$, ${}^1\alpha_k$, . . . , so that Fig. 8-12*a* implements

$$\left.\begin{aligned} {}^rF &= F({}^r\alpha_1, {}^r\alpha_2, \ldots, {}^r\alpha_n) \\ {}^r\alpha_i &= {}^{r-1}\alpha_i + {}^r\Delta\alpha_i \end{aligned}\right\} (r = 1,2, \ldots ; i = 1,2, \ldots ,n) \quad (8\text{-}71)$$

It remains to determine suitable corrections ${}^r\Delta\alpha_i$ for automatic optimization.

(c) Use of Parameter-influence Coefficients. Kokotovic Systems. By analogy with Eq. (28), iteration on the basis of

$${}^r\alpha_i = {}^{r-1}\alpha_i - h \left.\frac{\partial F}{\partial \alpha_i}\right]_{{}^{r-1}\alpha_1, {}^{r-1}\alpha_2, \ldots, {}^{r-1}\alpha_n}$$

$$(h > 0; r = 1,2, \ldots ; i = 1,2, \ldots ,n) \quad (8\text{-}72)$$

would approximate steepest-descent minimization of $F(\alpha_1,\alpha_2, \ldots ,\alpha_n)$ for a reasonably small *step-size factor h*. Since F is a functional, we cannot usually express $\partial F/\partial \alpha_i$ as an explicit function of the α_k. We may, however, be able to generate the $\partial F/\partial x_i$ as functions of parameter-influence coefficients $\partial y_k/\partial \alpha_i$ found through solution of sensitivity equations (Sec. 8-4).

As an example, consider the boundary-value problem of Eq. (63). Here $\dot{y}(0)$ is the only unknown parameter, and

$$\frac{\partial F}{\partial \dot{y}(0)} = \frac{\partial}{\partial \dot{y}(0)} [y(T) - y_T]^2 = 2[y(T) - y_T] \left.\frac{\partial y}{\partial \dot{y}(0)}\right]_{t=T} \quad (8\text{-}73)$$

where the parameter-influence coefficient $\partial y/\partial \dot{y}(0) = v(t)$ satisfies the sensitivity equation

$$\left.\begin{aligned} \frac{d^2v}{dt^2} &= -\omega^2 v \\ \text{with} \qquad v(0) &= 0 \\ \dot{v}(0) &= 1 \end{aligned}\right\} \quad (8\text{-}74)$$

obtained by differentiation of Eq. (63) with respect to $\dot{y}(0)$. We can, then, solve the sensitivity equation (74), along with Eq. (63), once per computer run; we sample $v(t)$ at $t = T$ to obtain $v(T)$, and implement Eq. (73) to produce $\partial F/\partial \dot{y}(0)$ (Fig. 8-14). Analogous procedures apply to our other examples. With multiparameter problems involving nonlinear differential equations, though, the sensitivity equations tend to be impractically cumbersome (or we may not even be able to write them in explicit form, Sec. 8-4).

Our simple example involved a single unknown parameter. In the general case of an nth-order system with r parameters, we should require an analog-computer setup for r nth-order sensitivity-equation systems (16) in addition to the original nth-order system. In practice, therefore, *explicit computation of parameter-influence coefficients is practical only for linear time-invariant systems of the so-called Kokotovic class,* which permits derivation of *all* parameter-influence coefficients from a *single* nth-order sensitivity-analysis setup.

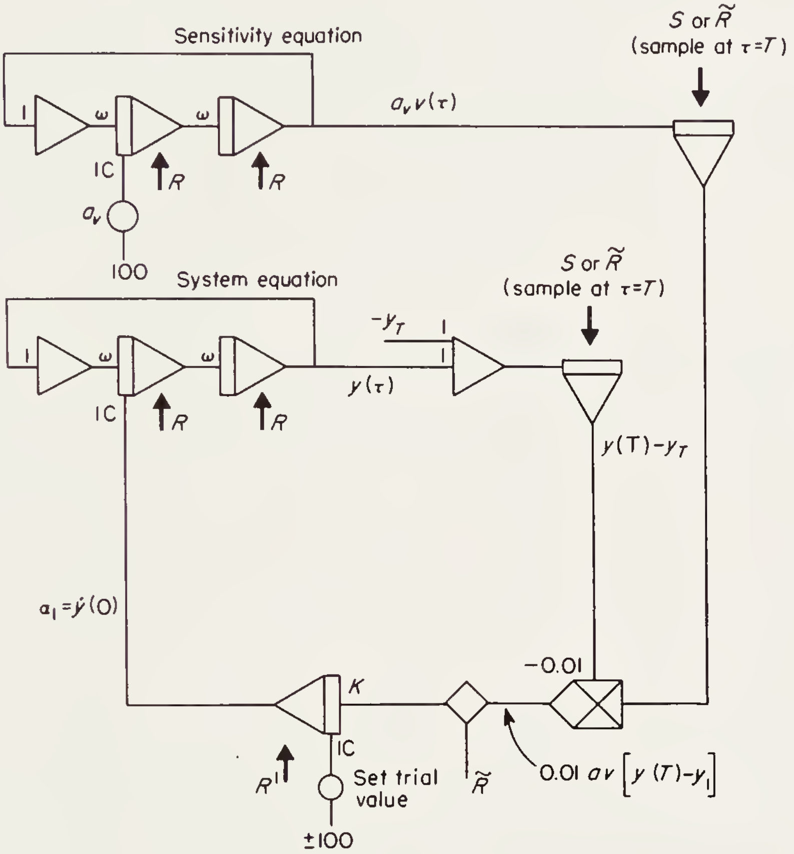


Fig. 8-14. Parameter optimization using a parameter-influence coefficient.

Kokotovic systems have the general form of Fig. 8-15*a*. The $H_k \equiv H_k(d/dt)$ are time-invariant linear operators of the form defined by Eq. (1-11), with $H_k(\infty)$ finite. For simplicity, assume that we want to determine sensitivity functions associated only with one unique parameter α_k corresponding to each H_k, so that $H_k \equiv H_k(d/dt, \alpha_k)$; this restriction can be relaxed. It can be shown that all r sensitivity

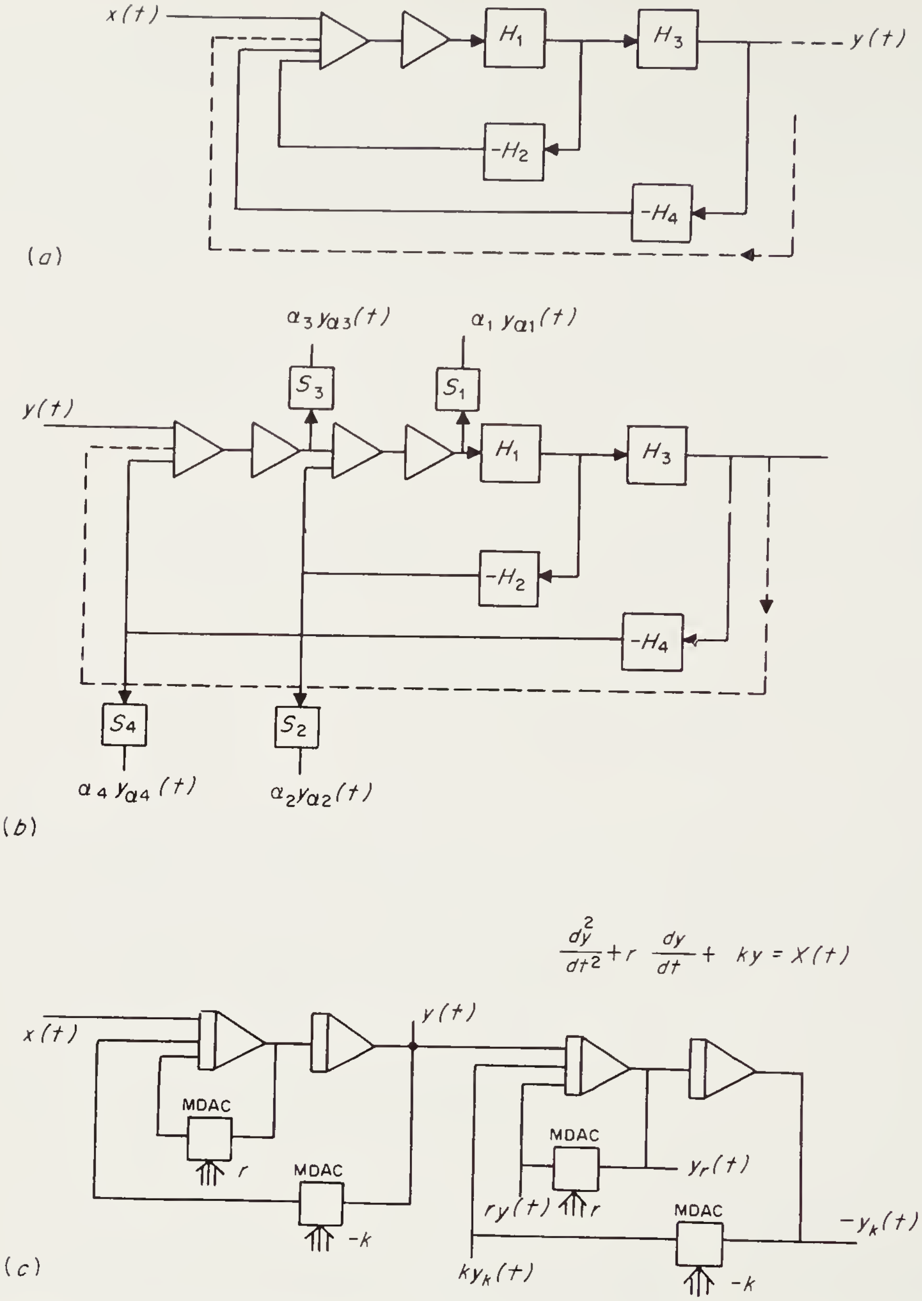


Fig. 8-15. A time-invariant linear system of the Kokotovic class (*a*), sensitivity-analysis model (*b*), and a simple example (*c*).

functions $\partial y/\partial\alpha_k \equiv y_{\alpha k}(t)$ of the system of *Fig.* 8-15*a can be obtained from the system shown in Fig.* 8-15*b whose input is the output* $y(t)$ *of the original system, and*

$$S_k\left(\frac{d}{dt}\right) \equiv \frac{\alpha_k}{H_k\left(\dfrac{d}{dt},\alpha_k\right)} \frac{\partial H_k\left(\dfrac{d}{dt},\alpha_k\right)}{\partial\alpha_k} \qquad (k = 1,2,\ldots)$$

The operators S_k are usually quite simple. Thus,

$$\text{If } H_k \equiv \alpha_k H\left(\frac{d}{dt}\right) \qquad \text{then } S_k \equiv 1$$

$$\text{If } H_k \equiv \frac{\alpha_k \dfrac{d}{dt}}{\alpha_k \dfrac{d}{dt} + 1} H\left(\frac{d}{dt}\right) \qquad \text{then } S_k \equiv \frac{1}{\alpha_k \dfrac{d}{dt} + 1}$$

Figure 8-15*c* shows a simple example.

(d) Computed-gradient Methods (Figs. 8-16 and 8-17). A more generally useful approach to steepest-descent minimization employs iteration based on

$$^{r}\alpha_i = {}^{r-1}\alpha_i - {}^{r-1}\left(\frac{\Delta F}{\Delta\alpha}\right)_i h \qquad (h > 0;\ r = 1,2,\ldots;\ i = 1,2,\ldots,n) \tag{8-75}$$

where the $^{r-1}(\Delta F/\Delta\alpha)_i$ are approximate gradient components *measured* and stored during n exploratory computer runs preceding each actual iteration step or "working step":

$$^{r-1}\left(\frac{\Delta F}{\Delta\alpha}\right)_i = \frac{1}{\Delta\alpha}[F({}^{r-1}\alpha_1, {}^{r-1}\alpha_2, \ldots, {}^{r-1}\alpha_i + \Delta\alpha, \ldots, {}^{r-1}\alpha_n) - F({}^{r-1}\alpha_1, {}^{r-1}\alpha_2, \ldots, {}^{r-1}\alpha_n)]$$

$$(r = 1,2,\ldots;\ i = 1,2,\ldots,n) \tag{8-76}$$

The basic gradient method requires $n + 1$ differential-analyzer runs per iteration step, and n track-hold memory circuits or storage registers switched by suitable subroutine counters will be needed to store the $^{r-1}(\Delta F/\Delta\alpha)_i$ during each cycle. Many useful modifications of the gradient methods have been designed, e.g., the *optimum gradient method* (Figs. 8-16 and 8-17*b*).[42–49] After finding a set of approximate gradient components (76), we do not remeasure the gradient at each step but continue with additional "working steps" based on the $^{r-1}(\Delta F/\Delta\alpha)_i$ *until F no longer decreases.* Only then we perform n exploratory runs to measure new gradient components.

(e) Termination of the Iteration. Step-size Determination. When an optimization routine no longer yields any improvement, we may:

1. Continue to let the parameter point oscillate about the optimum parameter combination, as in most optimizing control systems
2. Terminate the iteration and read the optimum parameter values, say as averages over the last few iteration steps
3. Terminate the iteration and change to a new optimization routine capable of producing more accurate results

The iteration may simply be terminated by an operator observing the process by reading F and/or $\alpha_1, \alpha_2, \ldots, \alpha_n$. For automatic termination and readout, we must add special circuits to detect, say, that all $\partial F/\partial \alpha_i$ have equal signs, or that recent iterations have not improved F. In particular, the maximum/minimum-reading circuit of Fig. 4-26 will not only store the highest or lowest value of F, but also produces a voltage step each time an improvement takes place. We terminate the iteration when a counter indicates that a preset number n_F of iteration steps have failed to improve F.

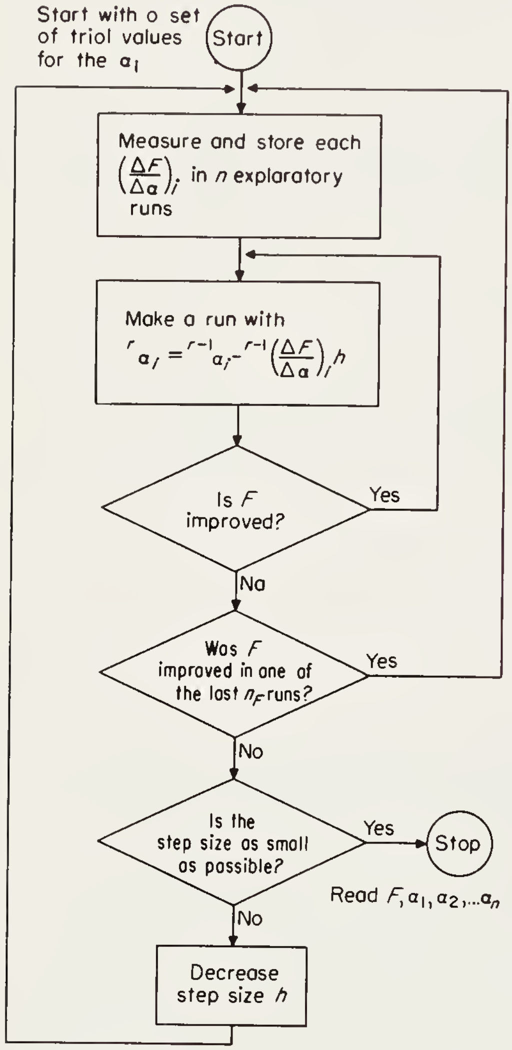
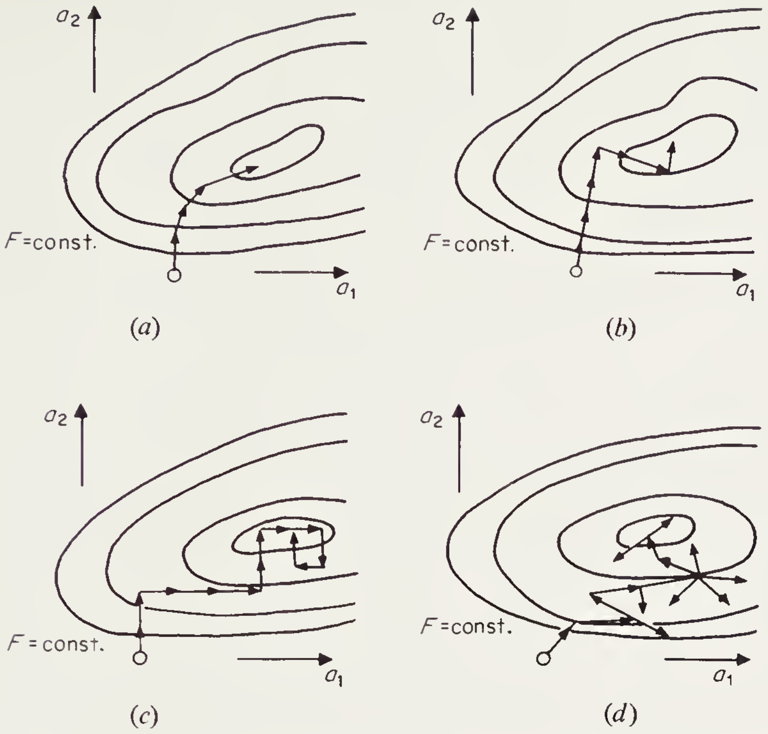

Start with a set of trial values for the a_i
Start
Measure and store each $\left(\frac{\Delta F}{\Delta a}\right)_i$ in n exploratory runs
Make a run with ${}^r a_i = {}^{r-1}a_i - {}^{r-1}\left(\frac{\Delta F}{\Delta a}\right)_i h$
Is F improved?
Yes
No
Was F improved in one of the last n_F runs?
Yes
No
Is the step size as small as possible?
Yes
Stop
Read $F, a_1, a_2, \ldots a_n$
No
Decrease step size h

Fig. 8-16. Flow diagram for iterative parameter optimization by the optimum gradient method.

In general, a small step-size factor h in Eqs. (72) and (75) will produce more accurate results but slower convergence to the desired optimum. Large steps may speed the iteration but might produce incorrect gradient measurements and large oscillations about optimal points. To combine speed and accuracy, it appears desirable to decrease the step size as we approach the desired optimum. With fast iterative differential analyzers, logic requirements will be simplified if we compromise on a reasonably small initial step size (say, 1 to 5 percent of half-scale) and reduce the step size once or twice as the search terminates. The hybrid parameter-setting circuit of Fig. 8-13*b* permits especially simple digital control of step size.

In certain optimization problems, we know a priori that the value of a minimum must be zero (as in the case of boundary-value problems, Fig. 8-12*a* and *b*) or near zero (as in many model-matching problems, Fig. 8-12*d*). In such cases, we may know that we are close to a minimum when F becomes small; we can then make the step size proportional to F, or to a monotonic function of F (see also Fig. 8-18*a*).



(a) (b) (c) (d)

Fig. 8-17*a* to *d*. Two-parameter optimization by the gradient method (*a*), by the optimum gradient method (*b*), by successive one-parameter optimizations (*c*), and by iteration with sequential random perturbations (*d*).

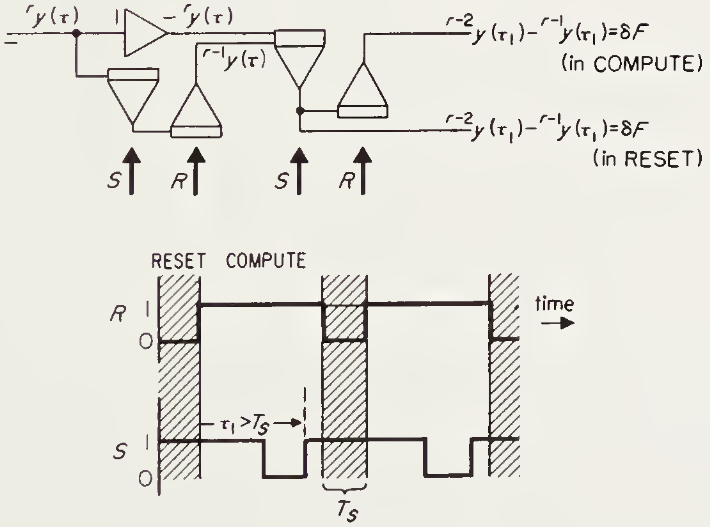


Fig. 8-17*e*. A circuit for presenting successive differences

$$\delta F = F({}^{\tau-2}\alpha_1, {}^{\tau-2}\alpha_2, \ldots, {}^{\tau-2}\alpha_n) - F({}^{\tau-1}\alpha_1, {}^{\tau-1}\alpha_2, \ldots, {}^{\tau-1}\alpha_n) = {}^{\tau-2}f(\tau_1) - {}^{\tau-1}f(\tau_1)$$

during the following COMPUTE period or during the following RESET period.

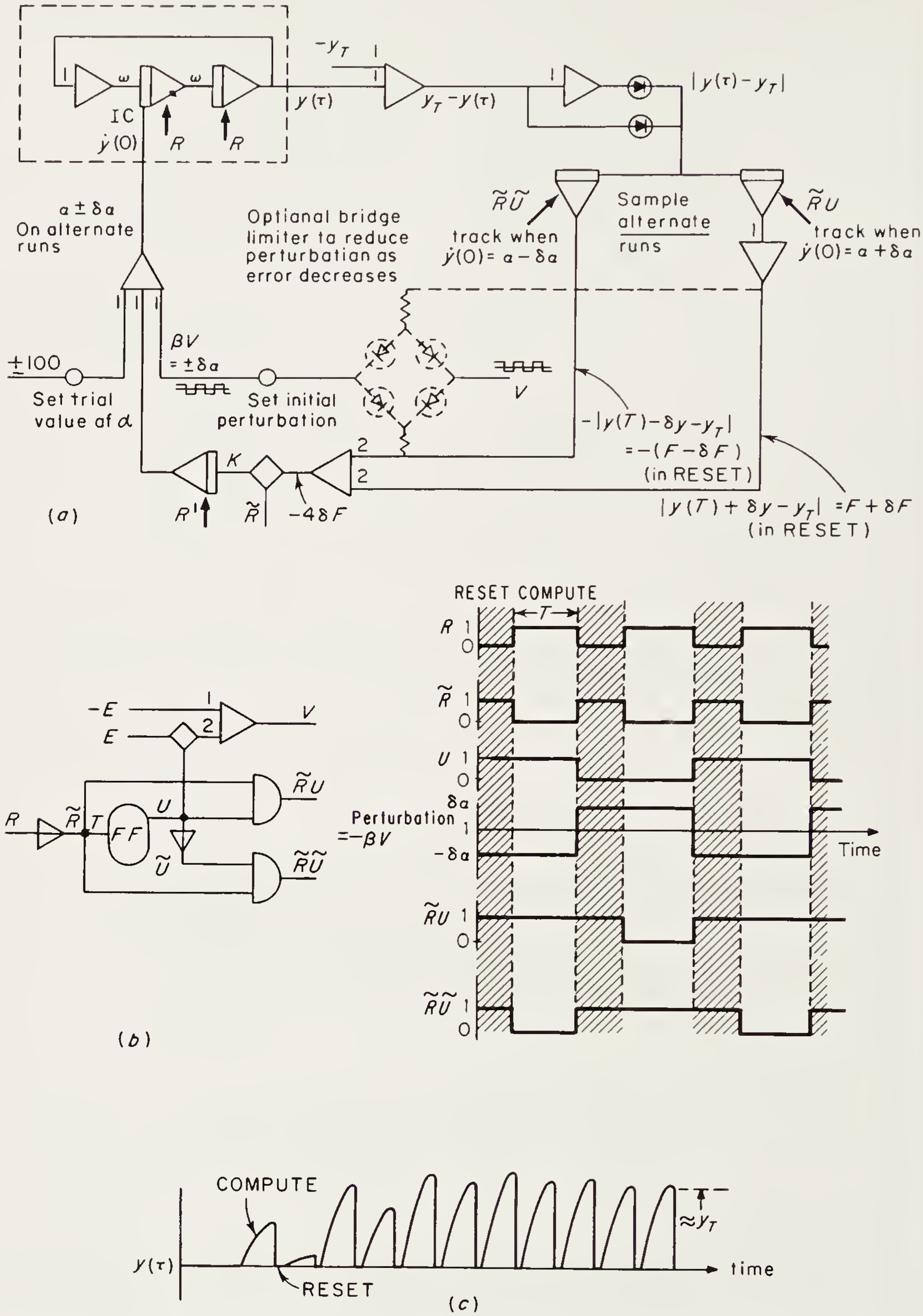


Fig. 8-18. A simple one-parameter parameter-perturbation optimizer (*a*), digital logic (*b*), and 10-Hz iterative solution of a simple boundary-value problem (*c*). For the boundary-value problem, the parameter $\alpha = y(0)$ is used only in RESET.

Even more rapid convergence to minima equal to or near zero may be obtained by a step-size factor h such that each new trial point $({}^r\alpha_1, {}^r\alpha_2, \ldots, {}^r\alpha_n)$ causes the linear approximation

$$F \approx F({}^{r-1}\alpha_1, {}^{r-1}\alpha_2, \ldots, {}^{r-1}\alpha_n) + \sum_{i=1}^{n} \frac{\partial F}{\partial \alpha_i} ({}^r\alpha_i - {}^{r-1}\alpha_i)$$

to vanish, where the derivatives are computed for ${}^{r-1}\alpha_1, {}^{r-1}\alpha_2, \ldots, {}^{r-1}\alpha_n$. We substitute ${}^r\alpha_i - {}^{r-1}\alpha_i$ from Eq. (72) to find[15]

$$h = h(r) = \frac{F({}^{r-1}\alpha_1, {}^{r-1}\alpha_2, \ldots, {}^{r-1}\alpha_n)}{\sum_{i=1}^{n} (\partial F/\partial \alpha_i)^2} \qquad (r = 1,2, \ldots) \tag{8-77}$$

and employ ${}^{r-1}(\Delta F/\Delta\alpha)_i$ instead of $\partial F/\partial\alpha_i$ when measured gradient components are used. Analog-computer implementation of Eq. (77) is rather complicated for our taste, but convergence may be rapid.

(f) Other Optimization Routines. One-parameter optimization is a relatively simple computing task (Fig. 8-18), while multiparameter optimization, say by a gradient method, requires a rather substantial amount of analog or digital memory and logic. With fast iterative analog computers, we might like to trade computing speed for simplified logic. Slow analog computers controlled by general-purpose digital computers, on the other hand, can employ more complicated optimization schemes involving higher-order extrapolation-prediction of maxima and minima and controlled step-size variation. References 42 to 46 survey the most interesting parameter-optimization methods; the *creeping-random-search method* will be singled out for special attention in Sec. 8-15, because it is especially well suited for hybrid-computer implementation.

A relatively simple multiparameter optimization routine optimizes F first as a function of a single parameter, say α_1. When the maximum or minimum is reached, we switch to the second parameter, optimize F again, and proceed until we start once more with the first parameter α_1 (Fig. 8-17*c*). Although this method will converge for reasonably smooth convex functions $F(\alpha_1,\alpha_2, \ldots, \alpha_n)$ and small step sizes, the iteration can "hang up" on two-dimensional ridges which maximize F in all parameter directions. Although such obstacles can be removed by supplementary search routines, this again detracts from the simplicity of the method.

8-15. Optimization by Creeping Random Search. Global Searches and Constraints.[56–75]

A completely different iteration method permits multiparameter optimization with relatively very simple digital programs or control logic, and not necessarily at the expense of convergence speed. Referring, for simplicity, to the two-parameter example of Fig. 8-17*d*, we start with a trial point $({}^0\alpha_1, {}^0\alpha_2)$ and vary α_1, α_2 simultaneously by independent *random* positive or negative increments $\delta\alpha_1$, $\delta\alpha_2$ obtained from a noise generator. If $F(\alpha_1,\alpha_2)$ is not improved, we try new random increments $\delta\alpha_1$, $\delta\alpha_2$ until an improvement is obtained; then we use $({}^0\alpha_1 + \delta\alpha_1, {}^0\alpha_2 + \delta\alpha_2)$ as the next trial point $({}^1\alpha_1, {}^1\alpha_2)$. With random perturbations distributed about zero with a small standard deviation (small step sizes), the iteration will converge whenever the gradient method does *but also in a great many cases where it does not.* Note that our random-perturbation scheme involves no exploration steps, varies all parameters simultaneously, and is much less sensitive to ugly parameter-space terrain features, such as ridges and canyons.

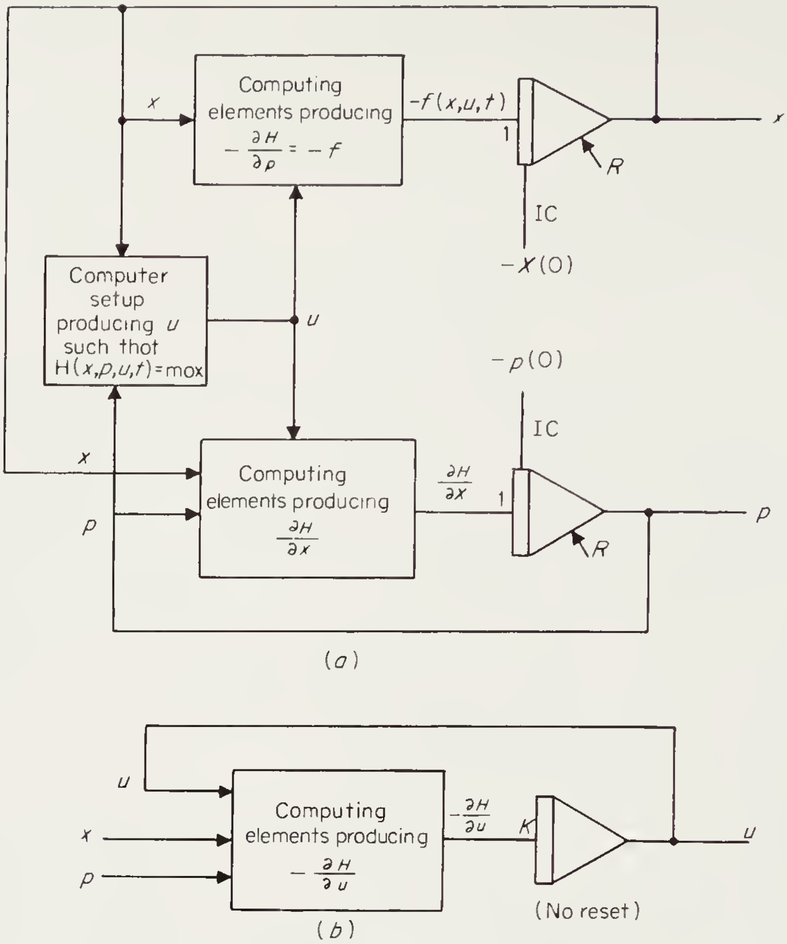

x
Computing elements producing $-\frac{\partial H}{\partial p} = -f$
$-f(x,u,t)$
1
R
x
IC
$-x(0)$
Computer setup producing u such that $H(x,p,u,t)$=max
u
$-p(0)$
IC
x
Computing elements producing $\frac{\partial H}{\partial x}$
$\frac{\partial H}{\partial x}$
1
p
R
p
(a)
u
x
p
Computing elements producing $-\frac{\partial H}{\partial u}$
$-\frac{\partial H}{\partial u}$
K
u
(No reset)
(b)

Fig. 8-19. Analog-computer implementation of Pontryagin's maximum principle (*a*), and steepest-descent computation of *u* as a function of *p* and *x* (*b*). (*Based on Ref.* 76.)

The increments $\Delta\alpha$ can be true random inputs from an analog gaussian-noise generator,[68,132] or uniformly distributed pseudorandom quantities from a shift-register pattern generator or digital-computer algorithm.[132]

Random-perturbation optimization is especially suitable for combinations of fast iterative analog computers and small digital computers. *Creeping random search really shines when the number of unknown parameters exceeds four.* In this case, both theory[63–66] and experiments[70] indicate that this optimization method converges more quickly than the basic measured-gradient method of Sec. 8-14*d* for simple convex criterion-function hypersurfaces. Indeed, the convergence time (for approach to the optimum within a specified small distance) was found to increase roughly *linearly* with the number R of unknown parameters, while competing methods exhibited exponential or square-law dependence.[70] Up to 15 parameters were successfully handled in fairly large problems,[68] but as many as 200 are possible.[77]

We have already noted that creeping random search can climb across multidimensional criterion-function landscapes with flat spots, winding canyons, ridges, etc. (Fig. 8-20*d*). In addition, parameter combinations producing analog-computer *overloads* are simply regarded as failures to improve F; the analog computer is reset before any amplifier really "blocks," and a new parameter perturbation is tried for the next run.

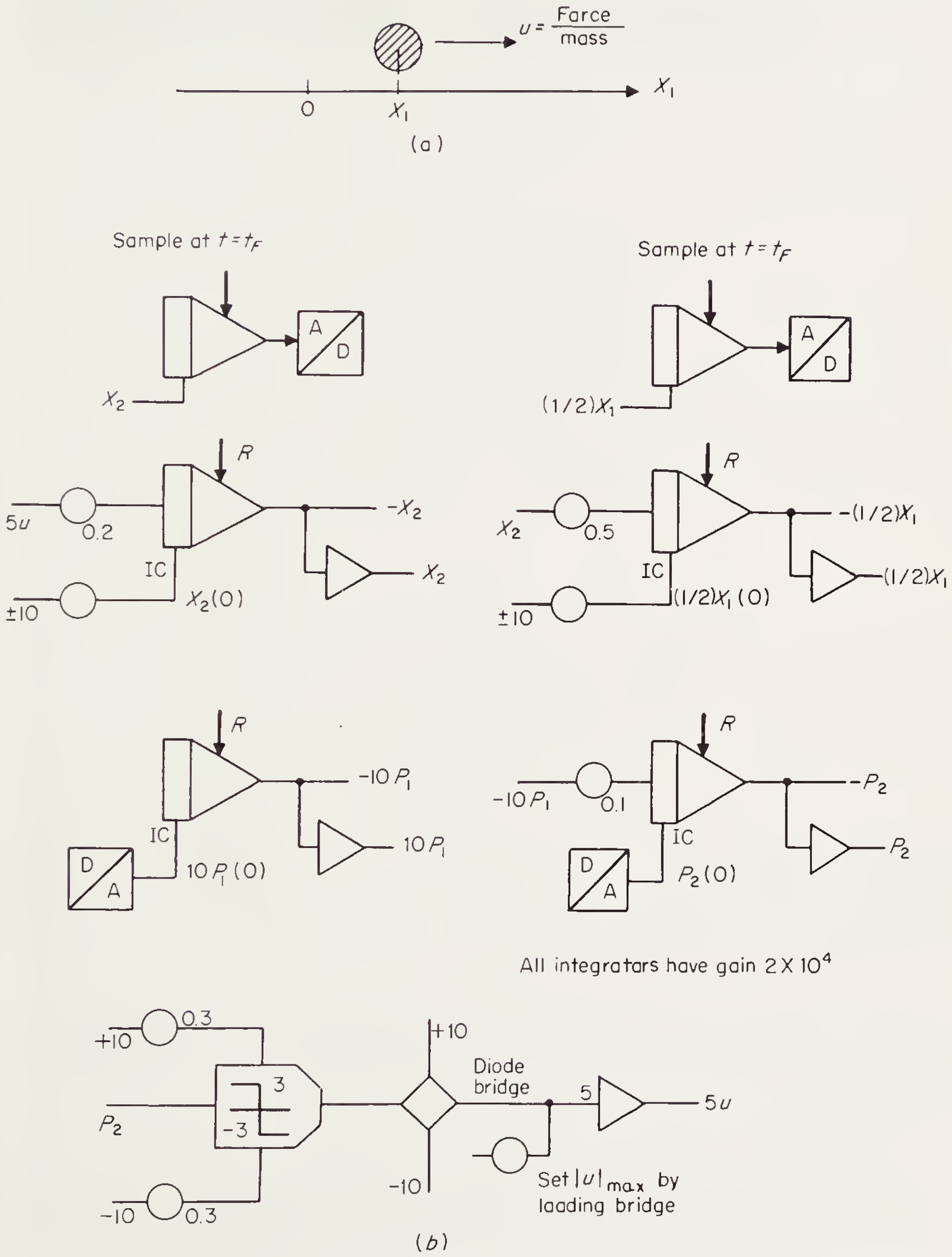


Fig. 8-20*a*, *b*. A simple bang-bang control system (*a*), and high-speed analog-computer setup (*b*) (see Sec. 8-16*b*). (*University of Arizona, Refs.* 67 *and* 73.)

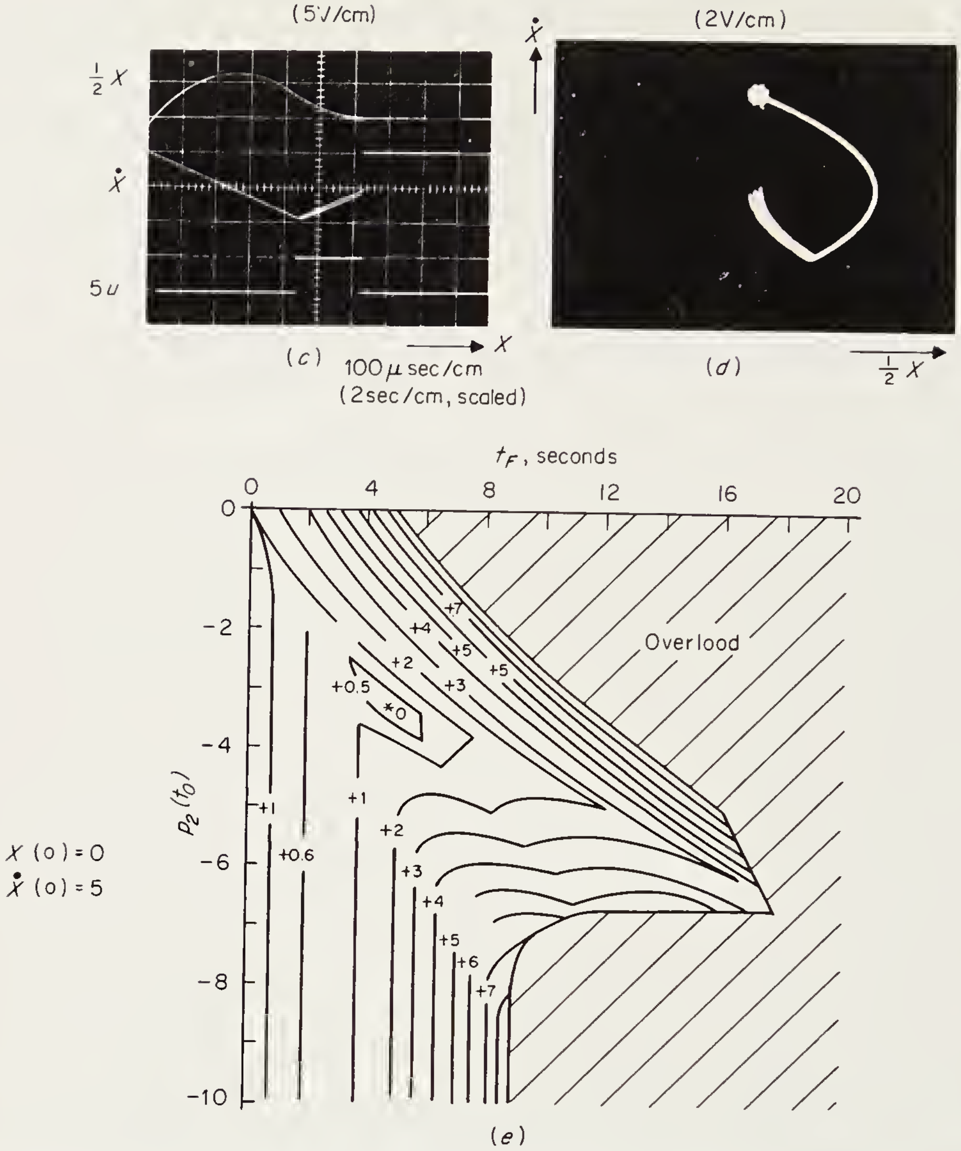


Fig. 8-20c to ***e***. Optimal solutions for the bang-bang control problem computed at 1,000 iterations/per second (*c*), phase-plane plot (*d*), and two-dimensional cross section of the criterion-function hypersurface (*e*). (*University of Arizona, Ref.* 67.)

Various *refinements* of the basic creeping random search can greatly improve convergence in *suitable* problem situations. The best strategy must be found by experimentation in each case.

1. As with gradient methods, the "step sizes," i.e., the perturbation variances, can and should be made adjustable. One good strategy is to *reduce* the variance after some number r_F of successive failures, so that we are less likely to "jump across" the optimum.[42,62,67]

2. After a specified still larger number of failures, we may substantially *increase* the perturbation variance in order to "jump out" of a flat spot, saddle, or local optimum.[62,67]
3. Another technique tries a larger *and* a smaller random perturbation in the same direction *at each step* and changes the variance so as to match the more favorable choice (adaptive random-step method of Schumer and Steiglitz).[70]
4. It is possible to utilize the experience of past successful or unsuccessful steps to give the search *preferred directions*, so that the creeping random search will begin to favor steepest-descent directions. We can (*a*) *repeat successful steps*,[67] (*b*) bias the mean of the perturbation in successful directions,[68] or (*c*) change the perturbation distribution by suppressing a specified proportion of positive or negative parameter increments.[62]

As a starting procedure, *large* parameter-perturbation variances may be used to approximate a *true random* search for a good starting point.

After an optimum has been found according to some termination criterion (Sec. 8-14*e*), we must decide, from a priori information or by experimentation, whether the optimum found is only a local optimum or the desired "global" optimum. In the latter case, a global search can be resumed with large trial steps.[62,67,73]

Equality constraints

$$\varphi_k(\alpha_1,\alpha_2,\ldots) = 0 \qquad (k = 1,2,\ldots,m)$$

on permissible combinations of the unknown parameters $\alpha_1, \alpha_2, \ldots$ can be handled by the *penalty-function technique* of Sec. 8-7; i.e., we simply optimize a modified criterion function in the manner of Eqs. (25) or (26); optimization of this new criterion function implies that the constraints are satisfied.

The penalty-function method can be used with all optimization algorithms. The newer *vector criterion function method* of Stewart et al., however, applies *only* to creeping random searches and will often be useful in this connection.[71–73] We consider an effective criterion function improved *only* if

$$F(\alpha_1,\alpha_2,\ldots),\ |\varphi_1(\alpha_1,\alpha_2,\ldots)|,\ |\varphi_2(\alpha_1,\alpha_2,\ldots)|,\ \ldots,$$

and $|\varphi_m(\alpha_1,\alpha_2,\ldots)|$ are *all* reduced by a new parameter perturbation.

Inequality constraints

$$\varphi_k(\alpha_1,\alpha_2,\ldots) \geq 0 \qquad (k = 1,2,\ldots)$$

can always be reduced to equality constraints as in Eq. (27).

8-16. True Functional Optimization. (a) Solution by Calculus of Variations and Maximum Principle. The problems of Sec. 8-13 required determination of optimal constant parameters $\alpha_1, \alpha_2, \ldots$. By contrast, true *functional* optimization requires us to find an unknown *function or functions* which will optimize a given functional. Consider, as an example, a simple control system with a control input $u(t)$ controlling a state variable $x(t)$ so that

$$\dot{x} \equiv \frac{dx}{dt} = f(x,u,t) \tag{8-78}$$

Given $x(0)$, we want to determine $u(t)$ so as to minimize the integral

$$x_0(t_F) \equiv \int_0^{t_F} f_0(x,u,t)\,dt \tag{8-79}$$

subject to the dynamic constraint (78). We shall employ the classical calculus of variations.[130] Although we might be able to use Eq. (78) for eliminating x in Eq. (79), a more general approach is to write Euler-Lagrange equations[130] with respect to both x and u for the "augmented" integrand

$$\Phi(x,\dot{x},u,t) \equiv f_0(x,u,t) + p(t)[\dot{x} - f(x,u,t)] \tag{8-80}$$

where $p(t)$ is a variable Lagrange multiplier.[130] The resulting necessary conditions for a minimum are two Euler-Lagrange equations

$$\frac{d}{dt}\frac{\partial\Phi}{\partial\dot{x}} - \frac{\partial\Phi}{\partial x} \equiv \frac{dp}{dt} - \frac{\partial f_0}{\partial x} + p\frac{\partial f}{\partial x} = 0 \tag{8-81}$$

$$\frac{d}{dt}\frac{\partial\Phi}{\partial\dot{u}} - \frac{\partial\Phi}{\partial u} \equiv p\frac{\partial f}{\partial u} - \frac{\partial f_0}{\partial u} = 0 \tag{8-82}$$

and a "natural" boundary condition[130]

$$\left.\frac{\partial\Phi}{\partial\dot{x}}\right]_{t=t_F} = p(t_F) = 0 \tag{8-83}$$

If the terminal time t_F is unknown (which is often true, as in Sec. 8-16*b*), the natural boundary condition (83) must be replaced by more general far-end boundary conditions (transversality conditions, Ref. 130).

Equation (82) implies that the hamiltonian function

$$H(x,p,u) \equiv pf(x,u,t) - f_0(x,u,t) \tag{8-84a}$$

satisfies $\partial H/\partial u = 0$ for optimal $x(t)$, $p(t)$, and $u(t)$, provided that $H(x,p,u)$ is suitably differentiable. Under more general conditions, *Pontryagin's maximum principle*[130] ensures that *the optimal control $u(t)$* continuously *maximizes the hamiltonian $H(x,p,u)$*. Assuming that an optimum control exists, the maximum condition will, in principle, permit us to express $u(t)$ in terms of the state variable $x(t)$ and the *adjoint variable* $p(t)$. These variables satisfy the differential equations (78) and (81), which can be rewritten as the pair of *canonical equations:*

$$\frac{dx}{dt} = \frac{\partial H}{\partial p} \qquad \text{(SYSTEM EQUATION)} \qquad (8\text{-}84b)$$

$$\frac{dp}{dt} = -\frac{\partial H}{\partial x} \qquad \text{(ADJOINT EQUATION)} \qquad (8\text{-}84c)$$

In addition, p and x must satisfy *initial and far-end boundary conditions;* it can also be shown that max $H = 0$ so that, in general, there is just enough information to determine all unknown integration constants and the (possibly) unknown terminal time t_F.[130]

Figure 8-19*a* shows the resulting computer setup. The control $u(t)$ is obtained from $x(t)$ and $p(t)$ through solution of $\partial H/\partial u = 0$ for $u(t)$, through explicit maximization of $H(x,p,u)$ (Sec. 8-16*b*), or by the steepest-descent method of Sec. 8-9 (Fig. 8-19*b*). *Our entire analysis applies to the general case of n state variables* $x_1, x_2, \ldots, x_n$, *n adjoint variables* $p_1, p_2, \ldots, p_n$, *and r control variables* $u_1, u_2, \ldots, u_r$ if we simply interpret x as an $n \times 1$ (column) matrix (*state vector*), p as an $1 \times n$ (row) matrix (*adjoint vector*), and u as an $r \times 1$ (column) matrix (*control vector*), so that $\partial H/\partial p$ is a column matrix and $\partial H/\partial x$ is a row matrix (see Ref. 130 for detailed specification of the boundary conditions and methods for handling constraints).

The maximum-principle approach reduces the optimum control problem to a two-point boundary-value problem involving 2n differential equations (84). This problem amounts to the parameter-optimization problem of finding enough unknown initial values $p_k(0)$ [and possibly also unknown $x_i(0)$] to match the remaining boundary conditions, i.e., *to minimize a boundary-error measure* in the manner of Fig. 8-12*b* or 8-18.

One redeeming feature is that the optimal boundary error is necessarily zero, so that *we have a convenient condition for stopping the iteration.* Otherwise, the two-point boundary-value problems arising from the maximum principle tend to be ugly:

1. The number of differential equations to be solved is *doubled.*
2. The adjoint variables p_k will rarely have any physical interpretation, so that they are *difficult to scale.* Moreover, the adjoint equations (84*c*) may be unstable.
3. More often than not, the boundary-error measure gives rise to a *completely wild criterion-function landscape* full of flat spots, ridges, canyons, and multiple minima (Fig. 8-20*d* and Refs. 71 to 73).

Nevertheless, a number of useful control problems have been solved on the basis of the maximum principle. The most successful approaches have employed creeping-random-search optimization of the boundary-error measure.[42,67,71–73,79,134,135]

As a *simple* example, let q represent an error in a control system described by

$$\frac{dq}{dt} = -q + u \equiv f(q,u,t) \tag{8-85}$$

We want to minimize a combination of error costs and control costs, say

$$x_0 \equiv \tfrac{1}{2}\int_0^{t_F} (q^2 + u^2)\,dt \tag{8-86}$$

Equation (82) implies $u = p$, and our boundary-value problem becomes

$$\left.\begin{aligned} &\frac{dq}{dt} = -q + p \qquad \frac{dp}{dt} = q + p \\ \text{with} \qquad &q(0) = q_0 \qquad\quad p(t_F) = 0 \end{aligned}\right\} \tag{8-87}$$

This is easy to solve with an iterative differential analyzer. A more complicated problem will be considered next.

(b) Simple Bang-Bang Time-optimal Control.[130] Given $x_1(0)$, $x_2(0) = \dot{x}_1(0)$, and the state equations

$$\frac{dx_1}{dt} = x_2 \qquad \frac{dx_2}{dt} = u(t) \tag{8-88a}$$

$\left[\text{i.e.,}\ \dfrac{d^2x_1}{dt^2} = u(t)\right]$, one desires to minimize the time

$$x_0(t_F) = t_F = \int_0^{t_F} dt \qquad (f_0 \equiv 1) \tag{8-88b}$$

required to reach the given terminal state $x_1(t_F) = x_2(t_F) = \dot{x}_1(t_F) = 0$ by choosing a control $u = u(t)$ such that

$$|u(t)| \le 1 \tag{8-88c}$$

Maximization of the hamiltonian

$$H \equiv p_1x_2 + p_2u - 1$$

subject to $|u| \le 1$ requires

$$u = \operatorname{sign} p_2 = \begin{cases} 1 & (p_2 > 0) \\ -1 & (p_2 < 0) \end{cases}$$

with

$$\frac{dp_1}{dt} = 0 \qquad \frac{dp_2}{dt} = -p_1 = \text{const}$$

so that $p_1 = p_1(0)$, $p_2 = p_2(0) - tp_1(0)$. The optimal trajectories in the x_1, x_2 plane (phase plane) are parabolic arcs corresponding to $u = 1$ and $u = -1$. These arcs intersect the "switching curve" corresponding to $p_2 = 0$, and each trajectory continues to the origin on the latter (Fig. 8-20*d*). Each specific trajectory depends on the unknown parameters $p_1(0)$, $p_2(0)$, which must be determined so that the given boundary conditions $x_1(t_F) = x_2(t_F) = 0$ are satisfied.

Figure 8-20 illustrates the iterative hybrid-computer solution of the bang-bang problem.[67,73] The analog-computer setup of Fig. 8-20*a* is run at 1,000 iterations per second. $p_1(0)$, $p_2(0)$, and t_F are the three unknown parameters to be optimized by creeping random search with the boundary-error criteria

$$|x_1(t_F)| = \text{Min} \qquad \text{and} \qquad |x_2(t_F)| = \text{Min}$$

About 2,500 iterations were needed on the average to obtain boundary errors within 0.1 volt of zero. Figure 8-20*c* and *d* show how the iterative solutions "hunt" for the optimum, and Fig. 8-20*e* illustrates the rather unfavorable criterion-function landscape.

(c) Direct Functional Optimization. *Direct methods of functional optimization*[130] approximately describe the unknown function (or functions) $u(t)$ in terms of N parameters $\alpha_1, \alpha_2, \ldots, \alpha_N$ and then optimize the criterion function *as in an ordinary N-parameter optimization problem;* constraints can be handled by either the penalty-function method or by the vector-criterion function method (Sec. 8-15).

An unknown function $u(t)$ might be approximated in the form

$$u(t) \equiv \alpha_1\Psi_1(t) + \alpha_2\Psi_2(t) + \cdots + \alpha_N\Psi_N(t)$$

where the $\Psi_k(t)$ are some conveniently chosen known functions. $u(t)$ can also be described in terms of a table of sample values $\alpha_k = u(t_k)$, from which $u(t)$ is reconstructed with or without interpolation or filtering; the sampling times t_k may or may not be uniformly spaced.

Since storage of many parameter or sample values poses no problem in hybrid computers, direct optimization methods may well replace the maximum-principle technique for many problems; note that direct optimization does not require us to solve and scale adjoint equations. The problem with direct optimization is that the number of unknown parameters needed to approximate a function $u(t)$ adequately may be quite large. One might want, say, 200 samples $u(t_k)$. The right way to deal with this problem will be to optimize a coarse approximation first and to increase the number of unknown parameters as the optimization proceeds.

Direct optimization of a control function $u(t)$ in terms of sample values $u(t_k)$ was successfully implemented first by the *measured-gradient method* of Sec. 8-14*d*.[78,133] With a large number of unknown parameters or sample values, creeping random search would seem to be capable of faster convergence (Sec. 8-15).

Figure 8-21 shows a creeping-random-search setup for optimizing a single control function $u(t)$ described by 200 evenly spaced samples $u(t_k) = u(k\,\Delta t)$ $(k = 0,1,2,\ldots)$.[77] Only one digital-to-analog converter is needed to feed successive $u(k\,\Delta t)$ to the analog computer, where analog-noise perturbations $\Delta u(t)$ are added. The new samples $u(k\,\Delta t) + \Delta u(k\,\Delta t)$ are read with an ADC and return to the digital computer. A pair of "swinging-buffer" arrays in core memory, directly addressed by an automatic data channel (Sec. 7-12), always store the *latest* $u(k\,\Delta t) + \Delta u(k\,\Delta t)$ and also the *last successful* $u(k\,\Delta t)$. Whenever the perturbations $\Delta u(t)$ improve the criterion function $f_0(t_F)$ sufficiently [i.e., if $\Delta f_0(t_F) < \epsilon$], then the buffer addresses (pointers) are swapped by the data-channel controller, and the most recent $u(k\,\Delta t) + \Delta u(k\,\Delta t)$ becomes the last successful array. The noise perturbations

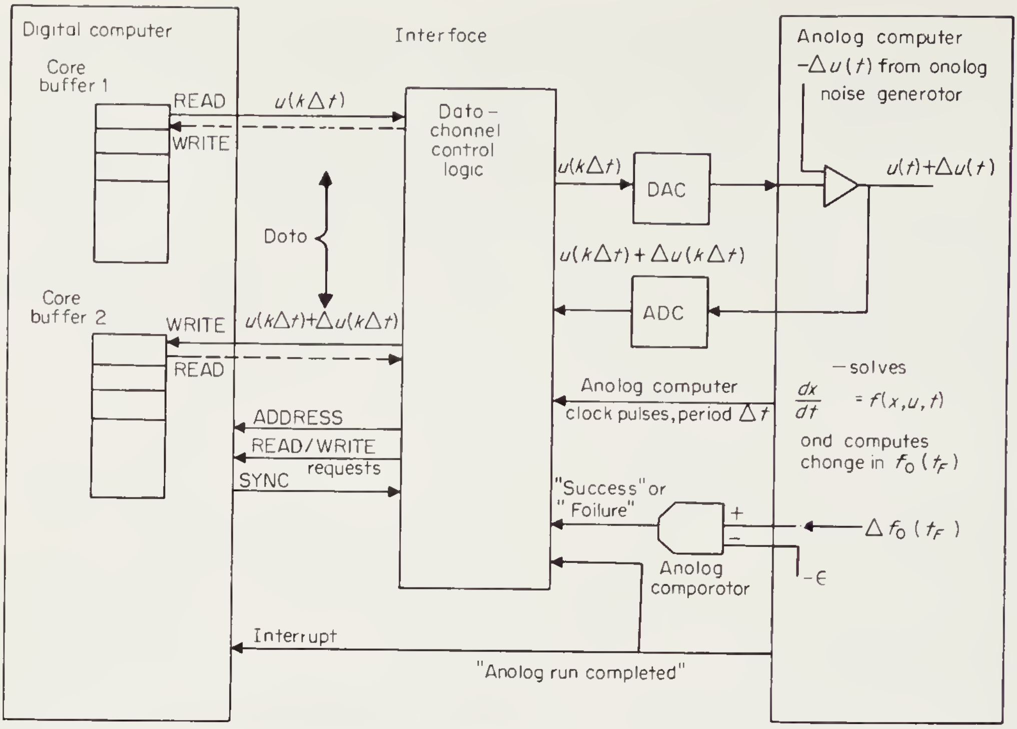


Fig. 8-21. Direct functional optimization. (*Based on Ref.* 77.)

can be modified in accordance with past failures and successes, as described in Sec. 8-15.

SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

8-17. Survey. Analog/hybrid-computer methods of solving partial differential equations require us to seek solutions in terms of ordinary differential equations, simple integrations, or, possibly, in terms of a multivariable equation-solving problem. Important applications include chemical process control, nuclear-reactor, and power-plant equations. This has become a fruitful area for new hybrid analog-digital computing techniques. To do justice to the mathematical sophistication of hybrid-computer solution of partial differential equations would require a book in its own right. We recommend the papers of R. Vichnevetsky[99–106] and especially his survey of the subject in Ref. 101.

SIMULATION OF SPECIAL CONTROL-SYSTEM NONLINEARITIES

8-18. Saturation, Limit Stops, and Backlash. A simple *saturation transfer characteristic* (Fig. 8-23*a*) approximating the behavior of various amplifiers, valves, motors, etc., is readily obtained with simple

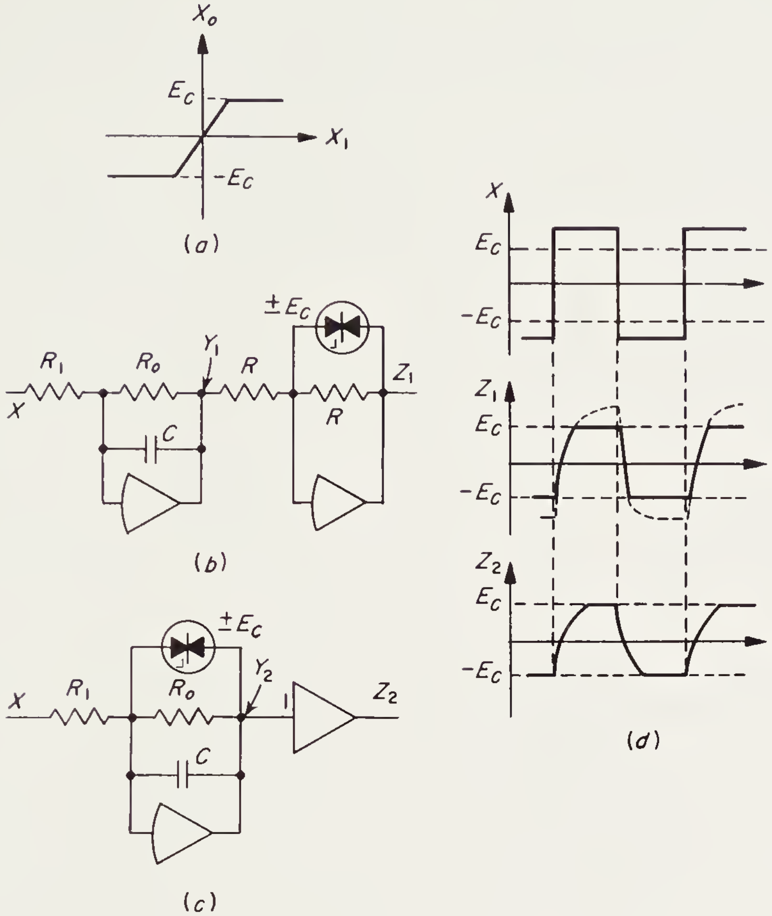


Fig. 8-22. Saturation characteristic (*a*), two types of saturation limiters preceded by a simple lag (*b*, *c*), and square-wave responses (*d*). The case $R_0 = \infty$ corresponds to an integration replacing the simple lag.

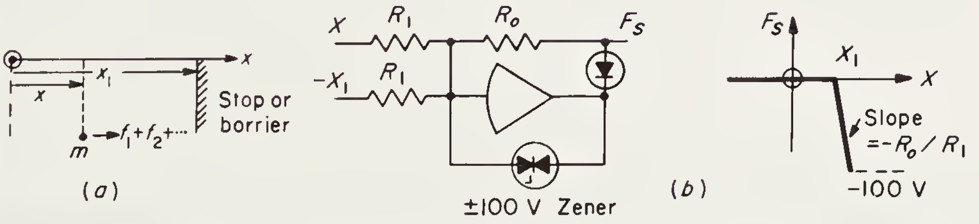


Fig. 8-23. Linear motion of a body restrained by a (fixed or movable) limit stop (*a*) and precision-limiter representation of an elastic stop with moderate stiffness (*b*).

shunt, feedback, or bridge limiter circuits (Secs. 4-3 and 4-6). Some care, however, is necessary if the simulated saturation characteristic is to be preceded by an integrator or time-lag circuit. Figure 8-22*b* and *c* illustrates two different possibilities; the behavior of such circuits must be carefully compared with the properties of the devices to be simulated in each particular case.

The action of a fixed *limit stop* on a moving particle body (Fig. 8-23) is *not* correctly represented by simple limiting of the particle displacement x. Direct simulation requires one to generate voltages proportional to the actual forces exerted by the limit stop as the moving particle penetrates the stop slightly.

These forces may be elastic in nature (*elastic stop*, energy is conserved); i.e., the force f_S exerted by the stop is given by

$$f_S = \begin{cases} -k_S(x - x_1) & (x > x_1) \\ 0 & (x \le x_1) \end{cases} \tag{8-89}$$

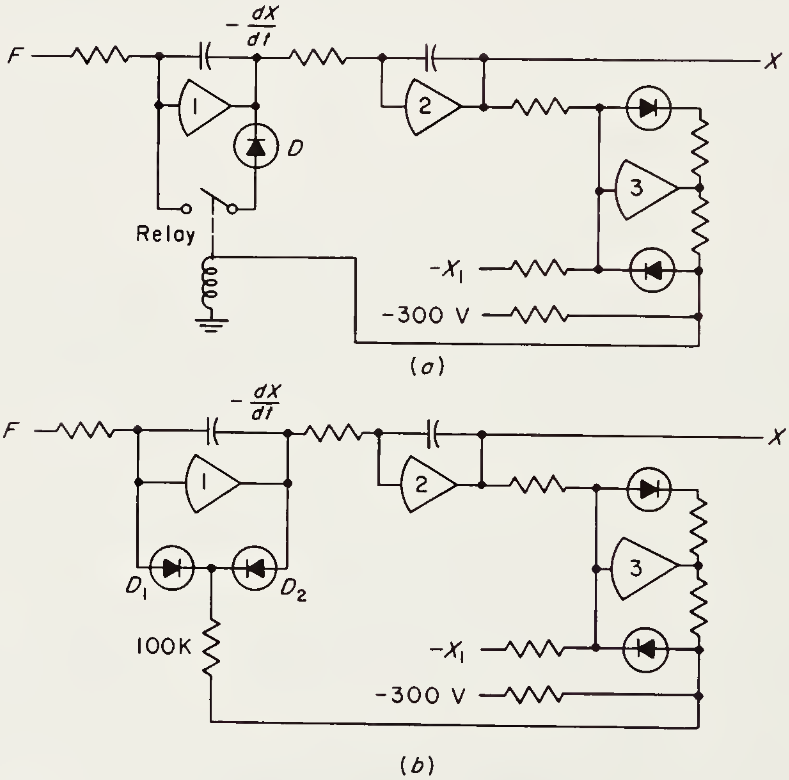


Fig. 8-24. Simulation of an inelastic stop with a relay (*a*) and with a diode switch (*b*). Diode D prevents "sticking." Note that the diode-bridge switch will limit the rate of discharge of integrator 1 (Sec. 5-2). In both circuits, finite diode forward resistances (50 to 100 ohms) permit the output of integrator 1 to differ slightly from zero for $F > 0$, so that some applications may require a second switch ahead of integrator 2 to prevent X from creeping.

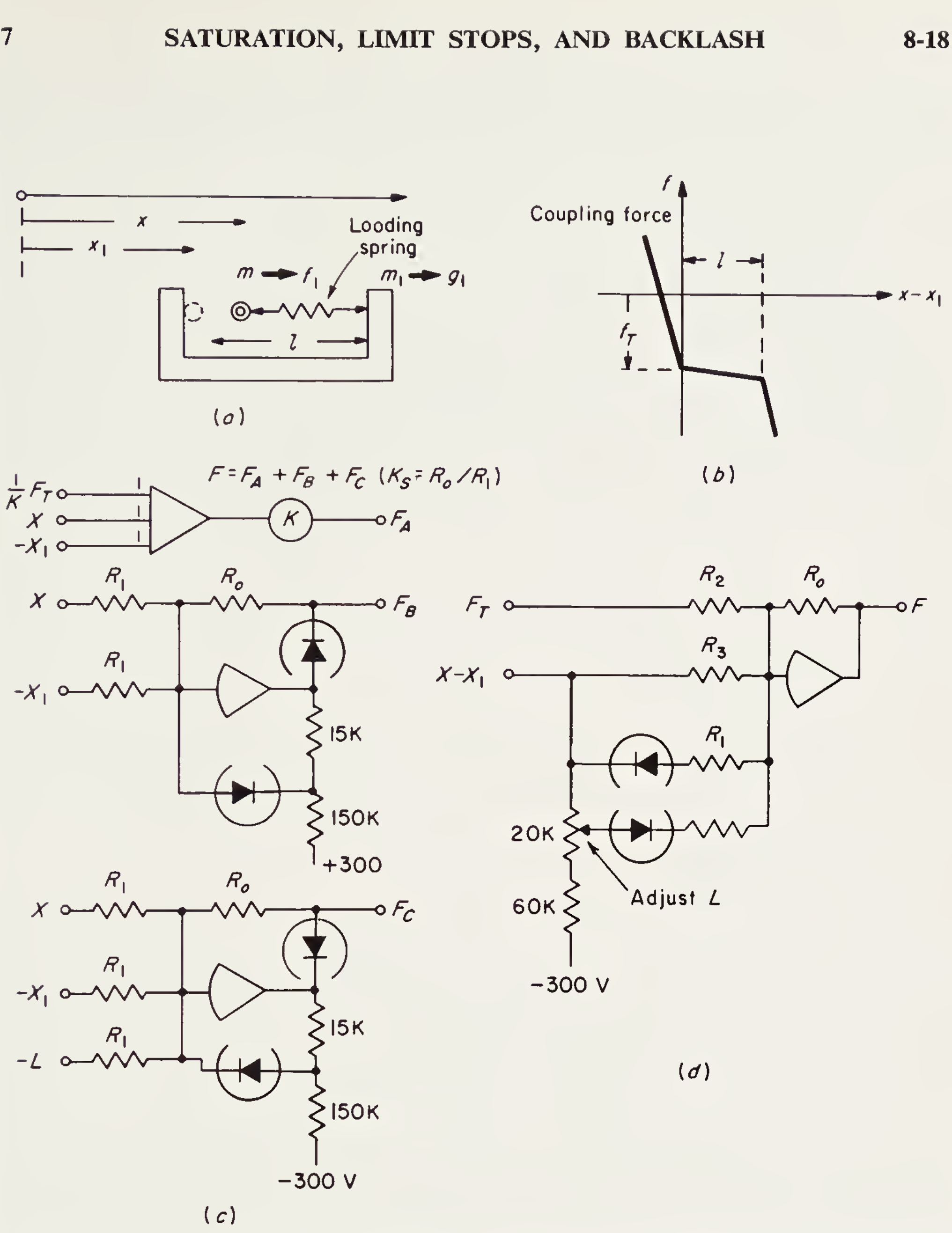


Fig. 8-25. Backlash in spring-loaded linkages or gears (*a*), coupling force on the driven member (*b*), and two diode circuits generating voltages proportional to the coupling force (*c*, *d*). The loading spring (antibacklash device) can be omitted. Note that an equal and opposite force acts on the driving member.

where $x - x_1$ is the penetration distance (Fig. 8-25), or the force f_S may be approximated by a viscous force

$$f_S = \begin{cases} -C_S \dfrac{d}{dt}(x - x_1) & \left[x > x_1, \dfrac{d}{dt}(x - x_1) > 0\right] \\ 0 \text{ otherwise} & \end{cases} \tag{8-90}$$

(viscous-type *inelastic stop*). If the force (90) were not zero for $(d/dt)(x - x_1) < 0$, the moving body would stick to the stop.[113] Combinations of such forces, or nonlinear forces depending on both $x - x_1$ and $\dot{x}$, result in *semielastic stops* with different coefficients of restitution, i.e., with partial dissipation of the initial kinetic energy. Figure 8-24 shows fabricated diode circuits for the simulation of inelastic stops.[112,113] If the moving body is limited to an interval between *two stops* ($x_1 \geq x \geq x_2$), the circuits of Figs. 8-23 and 8-24 may be duplicated with appropriately reversed diodes and bias voltages, or a deadspace circuit can be used. Many other variations and generalizations are possible. In particular, the stops themselves may be moving bodies, so that x_1 becomes a variable. The impact force on the stop will be equal and opposite to the force f_S; the computer can deal with motions involving impacts for two or more moving rigid bodies. In this manner, one can simulate *elastic backlash* (Fig. 8-25).

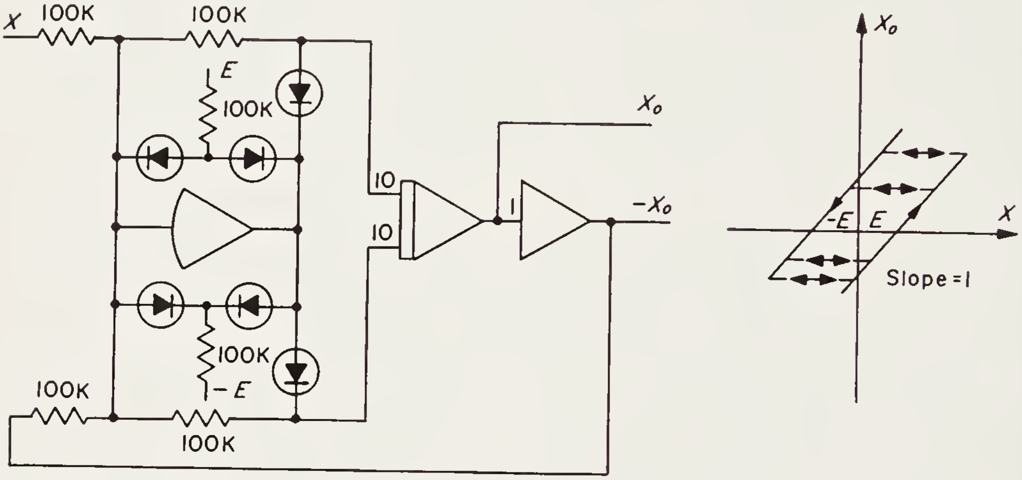


Fig. 8-26. An accurate "pseudobacklash" circuit employing the precision deadspace limiter of Fig. 4-22*c*. Other deadspace circuits can be substituted.

By contrast, the pseudobacklash circuit of Fig. 8-26 represents neither elastic impact nor a reaction force on the driving (input) member. This circuit is, however, useful for the simulation of backlash in potentiometers, process-control sensors, and other instruments which measure a process variable but do not affect the measured variable appreciably.

To simulate *oblique impacts* on smooth barriers, one must separately consider components of the impact forces in the coordinate directions (see also Sec. 8-10). It is, for instance, readily possible to simulate the rebound of a billiard ball from a barrier.

Simulation of hard elastic impact (e.g., steel on steel) requires caution, because the true impact forces can be so very large compared with all other forces in question that accurate scaling becomes quite impossible. In such cases, simulation circuits like those of Figs. 8-23 to 8-25 are necessarily inaccurate. Correct simulation of hard elastic, semielastic, or inelastic impacts requires *comparator-switched sample-hold-integrator circuits* which can store, reverse, and attenuate voltages proportional to each preimpact velocity component. A simulation circuit of this type is illustrated in Fig. 8-27.

8-19. Simulation of Static and Coulomb Friction. Dry friction acting, say, on a body capable of sliding in the x direction is commonly represented as a force

$$F_{\text{FRICTION}} = \left\{ \begin{array}{ll} -F_{\text{APPLIED}} & (|v| < \epsilon,\ |F_{\text{APPLIED}}| < F_S) \\ -F_S \operatorname{sign}(F_{\text{APPLIED}}) & (|v| < \epsilon,\ |F_{\text{APPLIED}}| \geq F_S) \\ -F_C \operatorname{sign}(v) & (|v| \geq \epsilon) \end{array} \right\} \tag{8-91}$$

where F_{APPLIED} is any external force applied in the x direction and $v = \dot{x}$ is the velocity of the sliding body. With the body at rest ($v = 0$), we have *static friction* equal and opposite to F_{APPLIED} until $|F_{\text{APPLIED}}|$ exceeds the breakaway force F_S. As the body begins to slide ($|v| > \epsilon$, where ϵ is a small velocity, say $\frac{1}{500}$ of the total velocity range), we have *coulomb friction*, i.e., a constant force of value $F_C < F_S$ opposing the velocity.

It will be noted that analog-computer simulation of dry friction requires generation of a function of *two* variables, since F_{APPLIED} depends on F_{FRICTION} as well as on v. McLeod[111] has implemented Eq. (91) with operational relays for slow real-time simulation; Fig. 8-28 shows an all-electronic circuit[120] believed to be substantially simpler as well as faster and more accurate than other diode circuits proposed for dry-friction simulation.[114–119] The circuit of Fig. 8-28 requires only two amplifiers in addition to three diode-bridge limiters (Sec. 4-6), which are available in many analog-computer patchbays. Phase inverter and diode-bridge limiter 1 cancel the applied-force voltage by adding $-F_{\text{APPLIED}}$ from the input of integrator 2 as long as $|v| < \epsilon$ and $F_{\text{APPLIED}} < F_S$. When F_{APPLIED} exceeds F_S, the bridge output is limited to F_S or $-F_S$, and

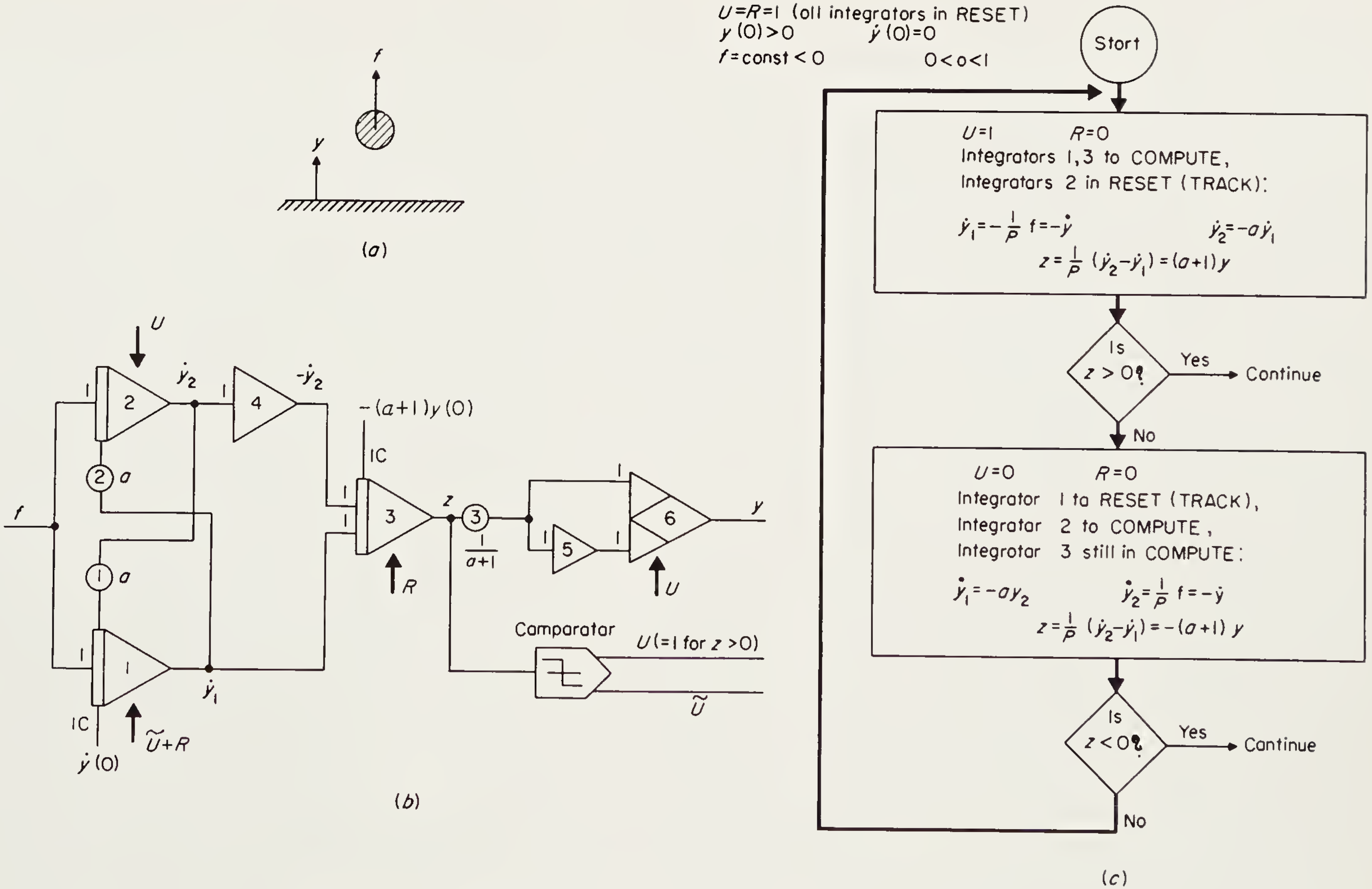
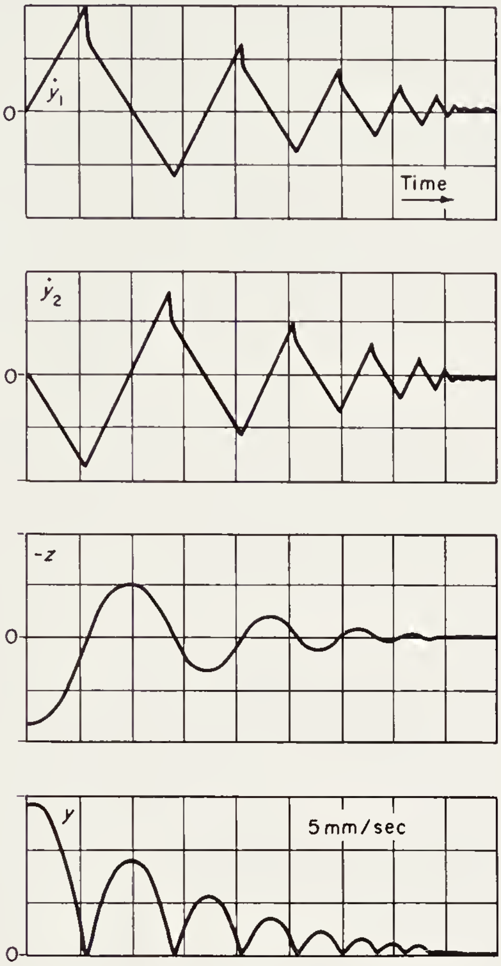

f
y
(a)
U
$\dot{y}_2$
$-\dot{y}_2$
1
2
4
② a
① a
1
3
5
6
IC
$-(a+1)y(0)$
z
③
$\frac{1}{a+1}$
R
U
y
Comparator
$U(=1 \text{ for } z>0)$
$\tilde{U}$
$\dot{y}_1$
IC
$\dot{y}(0)$
$\tilde{U}+R$
(b)
$U=R=1$ (all integrators in RESET)
$y(0)>0$ $\dot{y}(0)=0$
$f=\text{const}<0$ $0<a<1$
Start
$U=1$ $R=0$
Integrators 1,3 to COMPUTE,
Integrators 2 in RESET (TRACK):
$\dot{y}_1=-\frac{1}{p}f=-\dot{y}$ $\dot{y}_2=-a\dot{y}_1$
$z=\frac{1}{p}(\dot{y}_2-\dot{y}_1)=(a+1)y$
Is $z>0$?
Yes
Continue
No
$U=0$ $R=0$
Integrator 1 to RESET (TRACK),
Integrator 2 to COMPUTE,
Integrator 3 still in COMPUTE:
$\dot{y}_1=-ay_2$ $\dot{y}_2=\frac{1}{p}f=-\dot{y}$
$z=\frac{1}{p}(\dot{y}_2-\dot{y}_1)=-(a+1)y$
Is $z<0$?
Yes
Continue
No
(c)

Fig. 8-27. Semielastic impact (*a*), computer setup (*b*), and flow diagram (*c*).



(d)

Fig. 8-27*d*. Solution records obtained with the semielastic-impact simulation. In Fig. 8-27*a*, a steel ball impinges on a steel plate with velocity $-y$ and rebounds practically instantaneously with velocity *ay*; the coefficient of restitution α is between 0 and 1. The impact force is usually too large for satisfactory analog scaling (Sec. 8-18). Our iterative-differential-analyzer setup treats the period after each impact as a new "subroutine"; the new initial velocity *ay* is obtained through analog storage (Fig. 8-27*b*, *c*).

the resulting net integrator input can produce a velocity output in accordance with

$$\ddot{x} = \dot{v} = F_{\text{APPLIED}} - F_S \operatorname{sign} (F)_{\text{APPLIED}} \tag{8-92}$$

where a body of unit mass is assumed for simplicity.

Amplifier 3 serves as a deadspace comparator (Sec. 4-10), whose output equals zero when feedback diode bridge 2 conducts for $|v| < \epsilon$, but equals a sign (v) for $|v| \geq \epsilon$. The diode-bridge limiter 3 at the

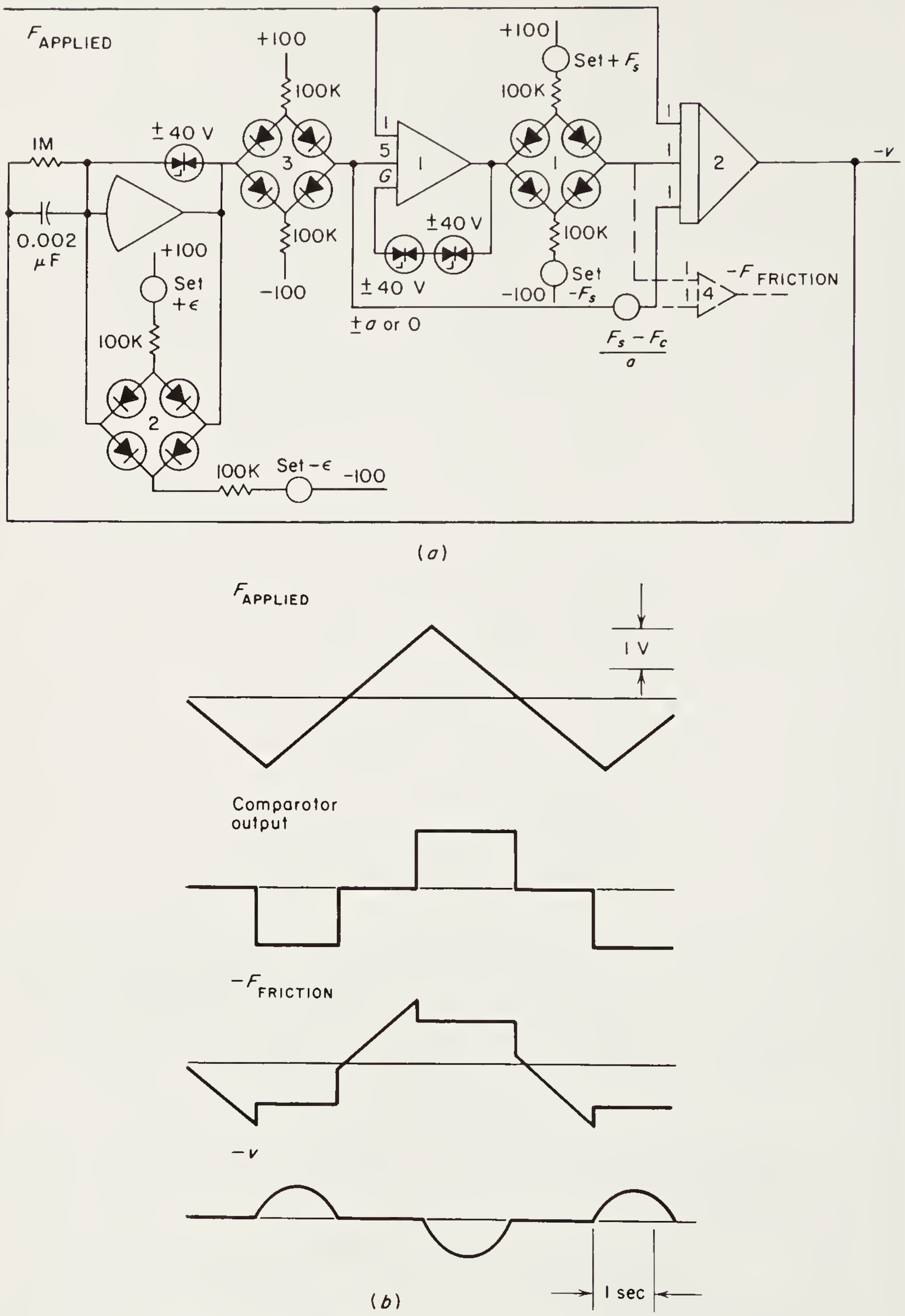


Fig. 8-28. Simulation of static and coulomb friction.

comparator output sets accurate limits $\pm a$ and also clips any comparator ringing. The comparator output is applied to amplifier 1 and drives the output of bridge 1 decisively into its limit $-F_S$ sign (v) whenever $|v| > \epsilon$. Since the required coulomb-friction force is $-F_C$ sign (v), a portion $(F_C - F_S)$ sign (v) of the comparator output is added to the integrator input. Amplifier 4 is not part of the computer setup proper but was added to display and record the total friction force.

Figure 8-28*b* shows results obtained using a 1-μF integrating capacitor; the transition between static and coulomb friction is clearly visible as the simulated body starts to move. Essentially similar results were obtained with a repetitive analog computer operating on a 1:1,000 time scale (0.001-μF integrating capacitor).[120]

8-20. Other Common Nonlinearities. Figure 8-29 illustrates simple representations of *relays* and *dry-friction clutches.* Figure 8-30 shows

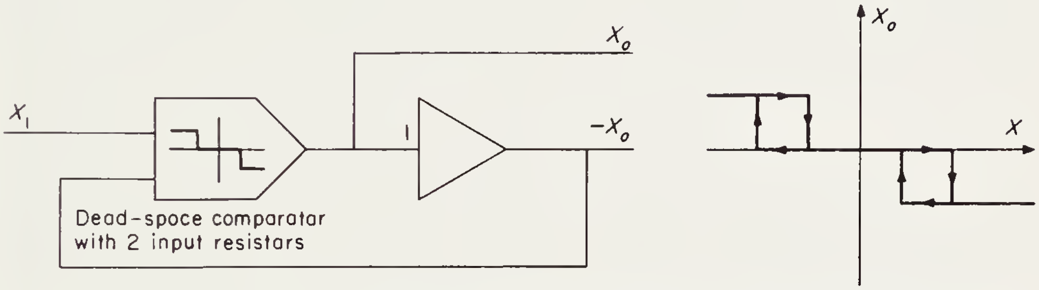


Fig. 8-29*a*. Deadspace/hysteresis circuit useful for simulation of relays, solenoids, clutches, and valves. The deadspace comparator of Fig. 4-22*b* is used.

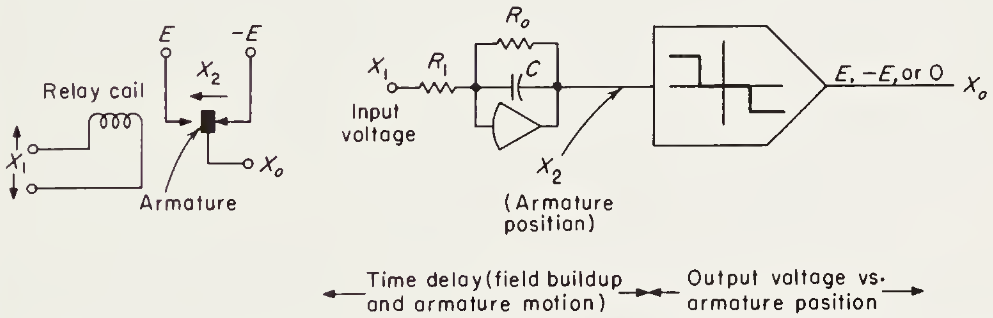


Fig. 8-29*b*. Another simplified representation of polarized double-throw relay. The delays involved in field-current buildup and armature motion are approximated by the transfer function

$$\frac{X_2}{X_1} = \frac{R_0}{R_1}\,\frac{1}{R_0CP + 1}$$

which is not unreasonable for well-designed relay circuits and sufficient armature damping. A more detailed simulation of the spring-restrained armature with damping and stops would be used for investigations of contact bouncing, effect of static friction, etc.

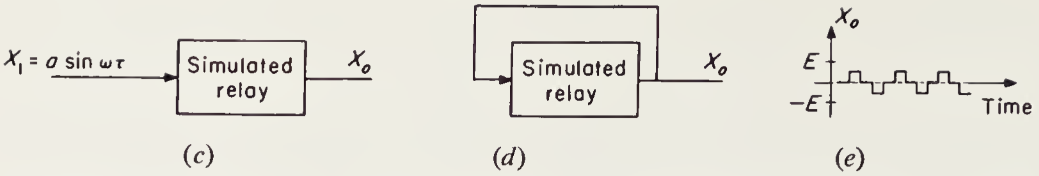


Fig. 8-29*c* to *e*. Generation of relay-chopper waveform with the circuit of Fig. 8-29*b*: externally driven relay (*a*), self-excited relay (buzzer) (*b*), and waveform (*c*).

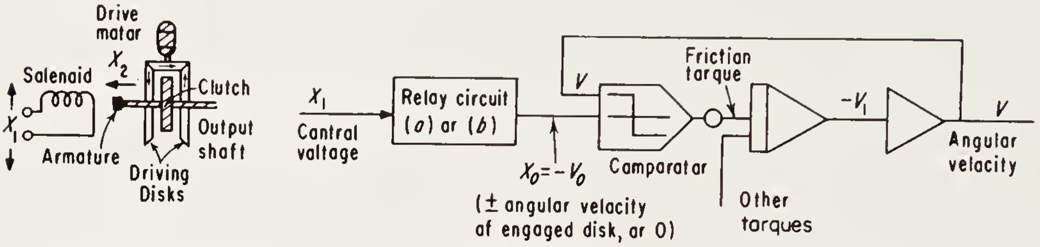


Fig. 8-29*f*. Simulation of a solenoid-operated dry-disk clutch. The dry-friction torque (see also Sec. 8-19) is developed from the difference between the output speed and speed of the driving disk engaged.

a circuit simulating *granularity* in wirewound potentiometers or other instruments.

A variety of *hysteresis-type transfer characteristics* can be obtained through combination of diode function generators with the bistable circuit of Fig. 4-26*a* or the pseudobacklash circuit of Fig. 8-26. Although such computer setups do exhibit hysteresis, they do not usually simulate the hysteresis characteristics of plastic and magnetic materials, steam-engine and compressor cycles, etc., correctly. Phenomena of this kind should not be simulated by trick diode circuits; correct hysteresis behavior must, instead, be reproduced by a computer setup implementing the differential equations governing the system dynamics.[123,124]

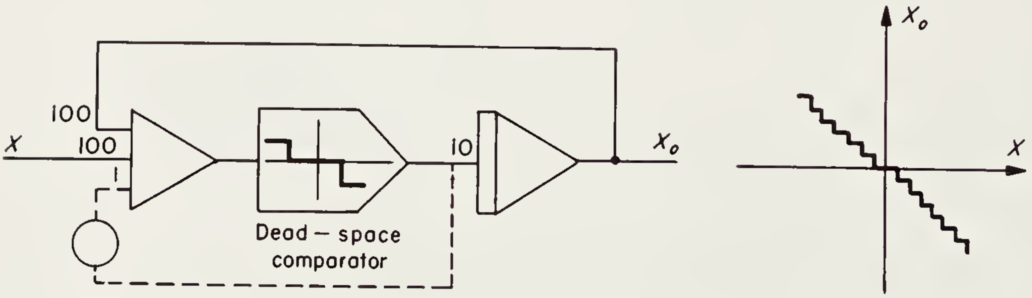


Fig. 8-30. Simulation of granularity in potentiometers and other control elements.

REFERENCES AND BIBLIOGRAPHY

Perturbation Methods, Parameter Sensitivity, Resubstitution Checks, and Automatic Rescaling

1. Miller, K. S., and F. J. Murray: The Mathematical Basis for the Error Analysis of Differential Analyzers, *MIT J. Math. Phys.*, nos. 2, 3, 1953.
2. Meissinger, H. F.: The Use of Parameter-influence Coefficients in Computer Analysis of Dynamic Systems, *Proc. Western Joint Computer Conf.*, 1960.
3. ———: Parameter-influence Coefficients and Weighting Functions Applied to Perturbation Analysis of Dynamic Systems, *Proc. 3d AICA Conf.*, Opatija, Yugoslavia, 1961, Presses Académiques Européennes, Brussels, 1962; see also *IEEETEC*, 1964.
4. Tomovič, R.: *Sensitivity Analysis of Dynamic Systems*, McGraw-Hill, New York, 1964.
5. Kokotovic, P.: Sensitivity Points Method, *Automatika i Telemekhanika*, December 1964.
6. ——— and S. Bingulac: Automatic Optimization of Linear Feedback Systems, *Ann. AICA*, January 1965.
7. Smith, D. A., and A. C. Soudack: Analog-computer Gradient Techniques for Parameter Optimization, *Ann. AICA*, October 1968.
8. Fogarty, L. E., and R. M. Howe: Analog-computer Solution of the Orbital Flight Equations, *IRETEC*, August 1962.
9. ———: Analog Simulation of the Re-entry of a Ballistic-missile Warhead and Multiple Decoys, *Proc. Spring Joint Computer Conf.*, 1962.
10. Howe, R. M.: High-speed Analog Simulation of Re-entry Trajectories in Three Dimensions, *Memo*, Applied Dynamics Corp., Ann Arbor, Mich., 1962.
11. Korn, G. A.: Parameter-perturbation Generator for Analog-computer Error and Optimization Studies, *Ann. AICA*, April 1963.
12. Meissinger, H. F.: Substitution Methods for the Verification of Analog Solutions, *Proc. Natl. Simulation Conf.*, Dallas, Tex., 1956; Simulation Councils, Inc., La Jolla, Calif.
13. Tomovič, R.: Picard's Method and Analog Computation, *IRETEC*, March 1957.
14. Matsuno, S.: Techniques for Obtaining Terminal Readout Accuracy in Homing-missile Simulations, *Memo.*, Hughes Aircraft Co., Culver City, Calif., 1959.
15. O'Grady, E. P.: Correlation Method for Computing Sensitivity Functions, *IEEETEC*, April 1967.
16. Gonzalez, R., and M. Muterspaugh: Optimization by Parameter-perturbation Correlation, *Simulation*, August 1969.
17. Ignatyev, M. B.: Control and Correction in Solving Differential Equations, *Proc. 5th AICA Conf.*, Lausanne, 1967.
18. ———: Checking and Correction of Analog and Hybrid Computation by the Redundant Variables Method, *Ann. AICA*, July 1968.
19. Sheehan, C. J.: An Analog-computer Simulation of the Restricted Threebody Problem by Automatic Scale-changing Techniques, *IEEETEC*, February 1964.
20. Kahne, S. J.: Automated Scaling for Hybrid Computers, *IEEETEC*, May 1969.

Solution of Algebraic Equations, Linear Programming, and Related Problems

21. Goldberg, E. A., and G. W. Brown: An Electronic Simultaneous-equation Solver, *J. Appl. Phys.*, **19**:339 (1948).
22. Marshall, B. O.: The Electronic Isograph for Roots of Polynomials, *J. Appl. Phys.*, **21**:307 (1950).
23. Pyne, I. B.: Linear Programming on an Analog Computer, *Communications and Electronics*, May 1956.
24. Neustadt, L. W.: A Method of Computing Eigenvectors and Eigenvalues on an Analog Computer, *Mathematical Tables and Other Aids to Computation*, **13**:194 (1959).

25. Neustadt, L. W.: Applications of Linear and Nonlinear Programming Techniques, *Proc. 3d AICA Conf.*, Opatija, Yugoslavia, 1961, Presses Académiques Européennes, Brussels, 1962.
26. Kovach, L. D., and H. F. Meissinger: Solution of Algebraic Equations, Linear Programming, and Parameter Optimization, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962 (note bibliography).
27. Rybashov, M. V.: Analog Solution of Algebraic and Transcendental Equations by the Gradient Method, *Automatika i Telemekhanika*, January 1961.
28. Korn, G. A.: Stabilization of Simultaneous-equation Solvers, *Proc. IRE*, **37**:1000 (1949).

Representation of Constraints and Related Steepest-descent Techniques

29. Greenwood, D. T.: The Representation of Constraints, *IRETEC*, September 1956.
30. Howe, R. M., and E. Gilbert: Trigonometric Resolution in Analog Computers by Means of Multiplier Elements, *IRETEC*, June 1957.
31. Corbett, J., et al.: Stabilization of Computer Circuits, *WADC Tech. Rept.* 57-425, Wright Air Development Center, November 1957.
32. Brunner, W.: Steepest Ascent, *PCC Rept.* 151, Electronic Associates, Inc., Princeton, N.J., 1959.
33. Skramstad, H. K.: Contour Tracing with an Analog Computer, *AIEE Winter General Meeting*, February 1959, *Paper* 59-461.
34. Turner, R. M.: On the Reduction of Error in Certain Analog-computer Calculations by the Use of Constraint Equations, *Proc. Western Joint Computer Conf.*, 1960.
35. Rubin, A.: Analog-computer Programming, Scaling, and Problem Preparation, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
36. Anderson, T. C.: Four-quadrant Division Circuits, *Inst. Control Systems*, April 1962.
37. Brunner, W., and R. Beadle: Amplitude Stabilization of Harmonic Oscillator with Diode Gates, *PCC Rept.* 173, Electronic Associates, Inc., Princeton, N.J., 1962.
38. Keller, R. E.: An Analog-computer Realization of the Euclidean Tools, *IRETEC*, August 1962; see also Ph.D. Thesis, Stanford University, 1962.
39. Talkin, A. I.: A Systematic Approach to Programming an Analog Computer to Generate a Large Class of Trajectories, *Harry Diamond Lab.*, *Memorandum*, U.S. Army Material Command, Washington, D.C., 1963.
40. Maybach, R. L.: Generation of Inverse Functions by the Method of Steepest Descent, *Ann. AICA*, October 1966.
41. Dastur, S. P., et al.: Analog/Hybrid Simulation Techniques for Linkage Synthesis, *Proc. SCSC*, Denver, Colo., 1970.

Iterative Hybrid-computer Parameter Optimization

42. Bekey, G. A., and W. J. Karplus: *Hybrid Computation*, Wiley, New York, 1968.
43. Balakrishnan, A. V., and L. W. Neustadt (eds.): *Computing Methods in Optimization Problems*, Academic, New York, 1964.
44. Wilde, D. J.: *Optimum-seeking Methods*, Prentice-Hall, Englewood Cliffs, N.J., 1964.
45. ——— and C. S. Beightler: *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, N.J., 1967.
46. McGhee, R. B.: Parameter Optimization Techniques, in M. Klerer and G. A. Korn, *Digital Computer User's Handbook*, McGraw-Hill, New York, 1967.
47. Feldbaum, A. A.: An Automatic Optimizer, *Automatika i Telemekhanika*, August 1958.
48. ———: Statistical Theory of the Gradient Method, *Automatika i Telemekhanika*, February 1960.
49. ——— and I. N. Bocharov: An Automatic Optimizer Which Searches for the Minimum of Several Minima (Global Optimizer), *Automatika i Telemekhanika*, March 1962.

50. Neustadt, L. W.: Synthesizing Time-optimal Control Systems, *J. Math. Anal. Appl.*, December 1960.
51. Moser, J. H., C. O. Reed, and H. L. Sellars: A Nonlinear Programming Technique for Analog Computation, Eastern Simulation Council Paper, December 1960, Shell Oil Co., Houston Research Lab., Deer Park, Tex.; see also *Instr. Control Systems*, March 1961.
52. Vichnevetsky, R. and J. P. Waha: Dynamic Programming and Hybrid Computation, *Ann. AICA*, April 1963.
53. Eckes, H. R.: "Digital Expansion System for ASTRAC I," M.S. Thesis, University of Arizona, 1963; *Ann. AICA*, January 1964.
54. Mitchell, B. A.: A Hybrid Analog-Digital One-parameter Optimizer, *ACL Memo* 69, University of Arizona, 1963.
55. Gilbert, E. G.: Hybrid-computer Solution of Time-optimal Problems, *Proc. Spring Joint Computer Conf.*, 1963.

Optimization by Random Perturbations

56. Brooks, S. H.: A Discussion of Random Methods for Seeking Maxima, *Operations Res.*, March–April 1958.
57. ———: A Comparison of Maximum-seeking Methods, *Operations Res.*, July–August 1959.
58. ——— and M. Ray Mickey: Optimum Estimation of Gradient Directions in Steepest-ascent Experiments, *Biometrics*, March 1961.
59. Favreau, R. R., and R. Franks: Random Optimization by Analog Techniques, *Proc. 2d AICA Conf.*, Strasbourg, France, 1958, Presses Académiques Européennes, Brussels, 1959.
60. Munson, J. K., and A. I. Rubin: Optimization by Random Search on the Analog Computer, *IRETEC*, June 1959.
61. Karnopp, D. C.: "Search Theory Applied to Parameter-scan Optimization," Ph.D. Thesis, Massachusetts Institute of Technology, 1961.
62. Mitchell, B. A.: "Hybrid Analog-Digital Parameter Optimizer for ASTRAC II," M.S. Thesis, University of Arizona, 1964; *Proc. Spring Joint Computer Conf.*, 1964.
63. Rastrigin, A. L.: Extremal Control by the Random-search Method, *Automatika i Telemekhanika*, September 1960.
64. ———: *Random Search in Optimization Problems* (in Russian), trans. by Air Force Systems Command, Foreign Technology Division, August 1967.
65. ——— and J. S. Rubinshtein: Comparison of Random Search and Stochastic Approximation, *Automatika i Vychisliteznaja Tekhnika*, September 1968.
66. Gurin, L. S., and L. A. Rastrigin: Convergence of the Random-search Method in the Presence of Noise, *Automatika i Telemekhanika*, **26**:1505 (1965).
67. Maybach, R. L.: Solution of Optimal-control Problems on a High-speed Analog Computer, *Simulation*, May 1966.
68. Bekey, G. A., et al.: Parameter Optimization by Random Search, *Proc. FJCC*, 1966.
69. Matyas, J.: Random Optimization, *Automatika i Telemekhanika*, February 1965; see also *Proc. 5th AICA Conf.*, Lausanne, 1967.
70. Schumer, M. A., and K. Steiglitz: Adaptive-step-size Random Search, *IEEETAC*, March 1968.
71. Kavanaugh, W. P., et al.: Optimal Control of Satellite Attitude Acquisition by a Random-search Algorithm, *Proc. SJCC*, 1968.
72. Stewart, E. C., et al.: Global Search Algorithm for Optimal Control, *Proc. 5th AICA Conf.*, Lausanne, 1970.
73. Gonzalez, R. S.: An Optimization Study on a Hybrid Computer, *Ann. AICA*, October 1970.
74. White, R. C.: "Hybrid-computer Optimization of Systems with Random Parameters," Ph.D. Thesis, University of Arizona, 1970; see also *Proc. 6th AICA Conf.*, Munich, 1970.
75. Janac, K.: Adaptive Stochastic Approximation, *Proc. SCSC*, Denver, Colo., 1970.

Functional Optimization

(See also Refs. 42 and 43)

76. Korn, G. A.: Steepest-descent Implementation of Pontryagin's Maximum Principle, *IEEETEC*, April 1964.
77. ——— and H. Kosako: A Proposed Computer Method for Functional Optimization, *IEEETEC*, February 1970.
78. Fogarty, L. E., and R. M. Howe: Trajectory Optimization by Direct Descent Process, *Simulation*, September 1968.
79. Miura, T., et al.: Hybrid-computer Solution of Optimal-control Problems by the Maximum Principle, *IEEETEC*, October 1967.

Model-matching Applications

80. Brunner, W.: An Iteration Procedure for Parametric Model Building and Boundary-value Problems, *Proc. Western Joint Computer Conf.*, 1961.
81. McGrath, R. J., and V. C. Rideout: A Simulator Study of a Two-parameter Adaptive System, *IRE Trans. PGAC*, February 1961.
82. McGrath, R. J., et al.: A Parameter-perturbation Adaptive Control System, IRE *Trans. PGAC*, May 1961.
83. Rajaraman, V.: Theory of a Two-parameter Adaptive Control System, *IRE Trans. PGAC*, July 1962.
84. Halbert, P. W.: Hybrid Simulation of an Aircraft Adaptive Control System, *Proc. Fall Joint Computer Conf.*, 1963.
85. Hofman, L. G., et al.: Theoretical and Experimental Research on Parameter-tracking Systems, *NASA Contractor Rept.* CR-452, Systems Technology, Inc., Hawthorne, Calif., 1966.
86. Eyckhoff, P.: Some Fundamental Aspects of Process-parameter Estimation, *IEEETAC*, October 1963.

Solution of Partial Differential Equations

87. Karplus, W. J.: *Analog Simulation: Solution of Field Problems*, McGraw-Hill, New York, 1958.
88. Clymer, A. B.: Operational-analog Simulation of the Vibration of Beams, *IRETEC*, September 1959.
89. ———: *Handbook of Equations and Operational-analog Circuits for the Solution of Structural Problems*, North American Aviation, Inc., Columbus, Ohio, 1959.
90. Liban, E.: The Application of Finite Fourier Transforms to Analog-computer Simulations, *Proc. Spring Joint Computer Conf.*, 1962.
91. Greenwood, D. T.: Network Analogies of Dynamical Systems, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962 (note bibliography).
92. Howe, R. M.: Solution of Partial Differential Equations, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
93. MacKay, D. M., and M. E. Fisher: *Analogue Computing at Ultra-high Speed*, Wiley, New York, 1962 (summarizes Fisher's earlier papers; note bibliography).
94. Giloi, W.: Über Stabilitätsschwierigkeiten bei der Lösung gewisser Fälle der Diffusionsgleichung, *Ann. AICA*, April 1963.
95. Miura, T., and J. Iwata: Time-sharing Computation Utilizing Analog Memory, *Ann. AICA*, July 1963.
96. Newman, D. J., and J. C. Strauss: Hybrid Assumed-mode Solution of Nonlinear Partial Differential Equations, *Proc. FJCC*, 1968.
97. Hara, H. H., and W. J. Karplus: Application of Functional Optimization Techniques for the Serial Hybrid-computer Solution of Partial Differential Equations, *Proc. FJCC*, 1968.
98. Presson, A. G., and H. J. Fink: Analog Relaxation Solution of the Nonlinear Ginzburg-Landau Equations, *Proc. AICA Conf. Hybrid Comp. Nucl. Eng.*, Versailles, 1968.
99. Vichnevetsky, R.: A New Method for Analyzing the Truncation Error in Finite-difference Representations, *Simulation*, March 1965.

100. Vichnevetsky, R.: Applications of Hybrid Computers to the Integration of Partial Differential Equations, *Proc. IFIP Congr.*, Edinburgh, 1968.
101. ———: Analog/Hybrid Solution of Partial Differential Equations in the Nuclear Industry, *Simulation*, December 1968.
102. ———: Generalized Finite-difference Approximations, *Simulation*, May 1969.
103. ———: Use of Functional Approximation Methods, *IEEETEC*, June 1969.
104. ———: Hybrid-computer Integration of Hyperbolic Partial Differential Equations, *Proc. 4th Australian Computer Conf.*, Adelaide, 1969.
105. ———: A Computing Method for the Simulation of Non-steady-state Burning of Solid Propellants, *Ann. AICA*, January 1970.
106. ——— and A. W. Tomalesky: A Hybrid-computer Method for the Analysis of Time-dependent River Pollution Problems, *Proc. SJCC*, 1970.
107. Chan, S. K.: Serial Solution of the Diffusion Equation Using Nonstandard Hybrid Techniques, *IEEETC*, September 1969.
108. Terasaki, R. M.: Analog Computation of Green's Function for Integrating Two-point Boundary-value Problems, *IRETEC*, February 1962.

Solution of Integral Equations

(See also Ref. 42)

109. MacKay, D. M., and M. E. Fisher: *Analogue Computing at Ultra-high Speed*, Wiley, New York, 1962 (summarizes Fisher's earlier papers; note bibliography).
110. Tomovič, R., and W. J. Karplus: *High-speed Analog Computers*, Wiley, New York.

Simulation of Control-system Nonlinearities

(See also bibliography for Chap. 4)

111. McLeod, J.: Electronic-analog-computer Techniques for the Design of Servo Systems, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
112. Korn, G. A., and T. M. Korn: Special Circuits for Dynamical-system Studies, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
113. Korn, G. A.: Simulation of Inelastic Stops, *IRETEC*, March 1960.
114. Monastyrshin, G. I.: Mathematical Simulation of Dry Friction, *Automatika i Telemekhanika*, December 1958.
115. Wang, H., and C. N. Shen: Simulating True Dry Friction, *Control Eng.*, October 1962.
116. Dunsmore, C. L.: Computer Analogs for Common Nonlinearities, *Control Eng.*, October 1959.
117. Langill A. W.: Accurate Simulation of Nine Common Nonlinearities, *Electron. Design*, June 21, 1961.
118. Zampino, J.: Southeastern Simulation Council Meeting, Oct. 20, 1961, reported in *Instr. Control Systems*, March 1962.
119. Terrazas, C., and B. B. Fannin: Simulating Discontinuous Nonlinear Functions, *Electron. Inds.*, August 1962.
120. Korn, G. A.: A New Analog-computer Setup for Simulation of Static and Coulomb Friction, *Instr. Control Systems*, September 1962.
121. Wong, Ming S.: Refraction Anomalies in Airborne Propagation, *Proc. IRE*, September 1958.
122. Howe, R. M.: Analog Techniques, *Instr. Control Systems*, August 1961.
123. Steingroewer, H.: Circuit for Hysteresis Loop, *PCC Rept.* 142, Electronic Associates, Inc., Princeton Computation Center, Princeton, N.J., 1959.
124. Simulation of a Reciprocating Compressor, *Application Bull.*, Electronic Associates, Inc., Long Branch, N.J.
125. Paynter, H. M.: Ordering and Selection Processes and Ultrareliable Systems, *AAAS paper*, December 1958; reprinted by G. A. Philbrick Researches, Inc., Boston, Mass.
126. Eckes, H. R.: Digital Expansion System for ASTRAC I, *Ann. AICA*, January 1964.
127. Brown, T. S., and K. F. Nelson: The Logical Simulation of Discontinuous Nonlinearities, *Simulation*, February 1966.

128. Thedchanamoorty, N.: Analog Quantizing Circuit, *Simulation*, November 1968.
129. Rubin, A. I., and G. F. Graber: Acoustic-ray Tracing on the General-purpose Electronic Analog Computer, *IEEETEC*, June 1965.

Miscellaneous

130. Korn, G. A., and T. M. Korn: *Mathematical Handbook for Scientists and Engineers*, 2d ed., McGraw-Hill, New York, 1968.
131. Bauer, L.: Aircraft, Autopilot, and Missile Problems, in H. D. Huskey and G. A. Korn, *Computer Handbook*, McGraw-Hill, New York, 1962.
132. Korn, G. A.: *Random-process Simulation and Measurements*, McGraw-Hill, New York, 1966.
133. Wingrove, R. C., and J. S. Raby: Trajectory Optimization Using Fast-time Repetitive Computation, *Proc. FJCC*, 1966.
134. Anderson, M. D., and S. C. Gupta: Backward-time Analog-computer Solution of Optimal-control Problems, *Proc. SJCC*, 1967.
135. De Backer, W.: The Maximum Principle, Its Computational Aspects, and Its Relation to Other Optimization Techniques, *EURATOM Rept*. 590e, EURATOM/CETIS, Ispra, Italy, 1964.

CHAPTER 9

HYBRID-COMPUTER ERRORS, SPECIAL TECHNIQUES, AND SOFTWARE REQUIREMENTS

INTRODUCTION

9-1. Survey. This chapter discusses some special problems and techniques encountered in hybrid analog-digital computation. Sections 9-2 to 9-6 describe *errors* introduced by the sampled-data nature of hybrid computations. Sections 9-7 and 9-8 to 9-9 respectively describe hybrid-computer *function storage* (including *time-delay simulation*) and *hybrid function generation*, two very important methods for supplementing analog simulation with digital techniques. The remainder of the chapter is concerned with the *digital programs and software systems* required for efficient hybrid-computer utilization.

ERRORS IN HYBRID COMPUTATION

9-2. Hybrid-computer Error Sources. In addition to errors inherent in the component analog and digital computers, hybrid computation produces new errors of its own, viz.:

1. *Gain and offset errors* in analog-digital-analog conversion, including such errors in sample-hold and multiplexer circuits. These errors can be treated as analog-computer errors.
2. *Timing and "skewing" errors* due to inaccurately timed and/or nonsimultaneous analog-digital or digital-analog conversion of different computer variables. Such errors can be made negligible with crystal-clock-timed analog sampling circuits and double-buffered DAC transfers.

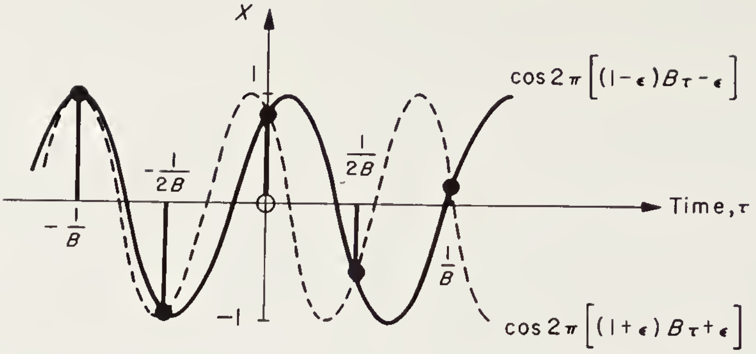

X
1
$\cos 2\pi[(1-\epsilon)B\tau-\epsilon]$
$-\frac{1}{2B}$
$\frac{1}{2B}$
$-\frac{1}{B}$
Time, τ
$\frac{1}{B}$
-1
$\cos 2\pi[(1+\epsilon)B\tau+\epsilon]$

Fig. 9-1. Folding-error demonstration. Sample values are identical for the two different functions shown.

3. *Quantization errors* due to the finite number of converted bits.
4. *Aliasing (folding) errors* cause faulty ADC outputs when the ADC input contains signal components faster than one-half the sampling frequency (Fig. 9-1). Filters designed to suppress such signal components can introduce phase-shift errors.
5. *Data-reconstruction errors* (Sec. 9-3).
6. *Errors due to the time delays inherent in digital processing and conversion* (Sec. 9-4).

In spite of a large literature on error analysis,[4–18] far and away the best method for estimating errors due to a specific source is to measure the effect of artificially increased errors (e.g., coarser quantization, greater time delay, faster time scale).

Quantization-error effects are rarely noticeable in larger hybrid computers having 14- to 15-bit conversion linkages, assuming that digital-computer roundoff errors have been avoided. Note that digital-computer integration will frequently require double-precision accumulation of integrals because of the large number of rounded terms added. In smaller hybrid systems, and especially in control and instrumentation systems, coarse quantization (8 to 12 bits, or even to 4 bits in statistical measurements[18]) can reduce costs; Refs. 16 to 18 present techniques for measuring and compensating quantization errors.

Data reconstruction and digital processing delays are usually by far the most serious error sources in closed-loop combined simulation, since they affect the stability of the hybrid-computer loop (Sec. 9-4).

One attempts to minimize the effects of the various error sources through *clever allocation of the analog and digital portions of a hybrid computation.* In general, the more quickly changing variables will be handled by the analog computer. *More sophisticated and accurate simulation techniques attempt to treat analog variables as (it is hoped small) corrections to digital variables, or vice versa*[5,57] (see also Secs. 8-1, 8-2, and 9-5).

9-3. Sampling and Data Reconstruction. A periodically operated sample-hold/ADC circuit with continuous input $X(\tau)$ produces a sampled-data sequence $X(0)$, $X(\Delta t)$, $X(2\Delta t)$, . . . , which represents a portion of the input information. The fundamental sampling theorem states that an analog signal without Fourier or spectral components beyond B Hz can be reconstructed exactly from $2B$ periodic samples per second, but practical circuits usually require at least 10 times this minimum sampling rate. Unequivocal conversion requires careful removal of any signal and/or noise components beyond B Hz by low-pass filters or averaging circuits[5] preceding the sampler to prevent *aliasing or folding errors* (Fig. 9-1).

The stepwise zero-order-hold output $Y(k\ \Delta t)$ from a typical double-buffered DAC represents the reconstructed analog variable correctly only at the beginning of each HOLD period; even then, it usually represents a delayed output (Fig. 9-2*a*). Simple smoothing with a low-pass filter may help but is sure to introduce phase shift. For better data reconstruction one can employ analog circuits to generate *extrapolation polynomials* matching successive data points $Y(k\ \Delta t) = Y_k$. The simplest example is the *first-order hold*, which implements linear extrapolation using the last two data points (linear *interpolation* between the last and the next new data point would be more accurate, but the next data point is in the future and can only be approximately predicted). rth-degree polynomial extrapolation yields exact reconstruction if the desired signal is a polynomial of degree $\leq r$ (Fig. 9-2 and Table 9-1).

Table 9-1 shows the *minimum sampling rates* needed by zero-order, first-order, and true polygonal hold circuits to reproduce a sinusoidal output with a specified percentage error. Note the striking improvement produced by a first-order-hold circuit. In spite of this, most hybrid-computer linkages simply implement zero-order holds.

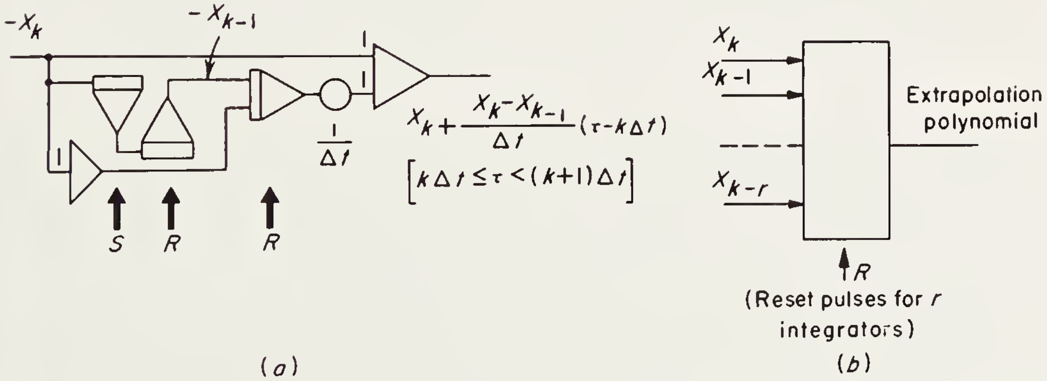
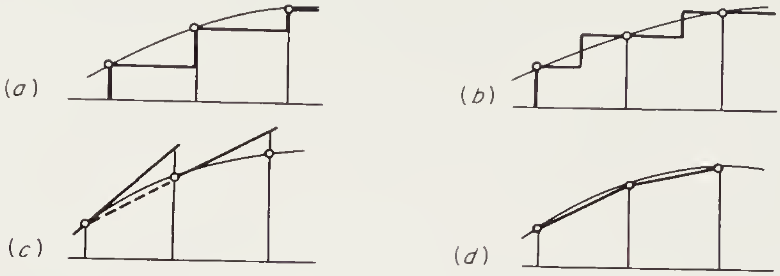

$-X_k$
$-X_{k-1}$
1
1
1
$\frac{1}{\Delta t}$
$X_k + \frac{X_k - X_{k-1}}{\Delta t}(\tau - k\Delta t)$
$[k\Delta t \leq \tau < (k+1)\Delta t]$
S
R
R
(a)
X_k
X_{k-1}
X_{k-r}
Extrapolation polynomial
R
(Reset pulses for r integrators)
(b)

Fig. 9-2. A first-order-hold circuit (*a*), and a more general hold circuit (*b*), for periodic sampling at the rate $1/\Delta t$.

Table 9-1. Data-reconstruction Errors for Sinusoidal Waveforms $\sin 2\pi ft$

Error formulas



(a) Zero-order-hold error
$$= \sin 2\pi f(t + \Delta t) - \sin 2\pi ft$$
$$\approx 2\pi f\,\Delta t \cos 2\pi ft \qquad [\le 200\pi f\,\Delta t \text{ percent}]$$

(b) Zero-order-hold error with data predicted $\Delta t/2$ sec ahead
$$= \sin 2\pi f(t + \Delta t) - \sin 2\pi f\left(t + \frac{\Delta t}{2}\right)$$
$$\approx \pi f\,\Delta t \cos 2\pi ft \qquad [\le 100\pi f\,\Delta t \text{ percent}]$$

(c) First-order-hold error
$$= 2 \sin 2\pi ft - \sin 2\pi f(t + \Delta t) - \sin 2\pi f(t - \Delta t)$$
$$\approx (2\pi f\,\Delta t)^2 \sin 2\pi ft \qquad [\le 100(2\pi f\,\Delta t)^2 \text{ percent}]$$

(d) First-order-polygonal-hold error
$$= \sin 2\pi f(t + \lambda) - \frac{\lambda}{\Delta t}[\sin 2\pi f(t + \Delta t) - \sin 2\pi ft] - \sin 2\pi ft$$
$$\approx \tfrac{1}{2}(2\pi f)^2(\lambda\,\Delta t - \lambda^2) \sin 2\pi ft \qquad (0 < \lambda < \Delta t)$$
$$\left[\le 100\,\frac{(2\pi f\,\Delta t)^2}{8} \text{ percent}\right]$$

Number $(1/f\,\Delta t)$ of sine-wave samples per period needed for a given percentage error

$1/f\,\Delta t$	Zero-order hold*	First-order hold	Polygonal hold
10	62.8	20	7
1	628	63	22
0.1	6,280	200	70
0.01	62,800	628	220

* Divide by 2 if data are predicted $\Delta t/2$ sec ahead.

The especially simple first-order-hold circuit of Fig. 9-3 is an improved version of an earlier circuit developed by L. Lofgren.[1,2] The circuit equation in HOLD (switch OFF) is

$$R_1C_1\frac{di}{d\tau} + i = aC_1\frac{dY_0}{d\tau} \qquad \text{with} \qquad i = C\frac{dY_0}{d\tau}$$

For first-order-hold action, $dY_0/d\tau$ must be constant in HOLD, so that $di/d\tau = 0$, and $a = C/C_1$. With C and C_1 initially discharged, let C be charged quickly to ΔY_0 during the first TRACK interval. First-order-hold action requires that the initial current $-a\,\Delta Y_0/R_1$

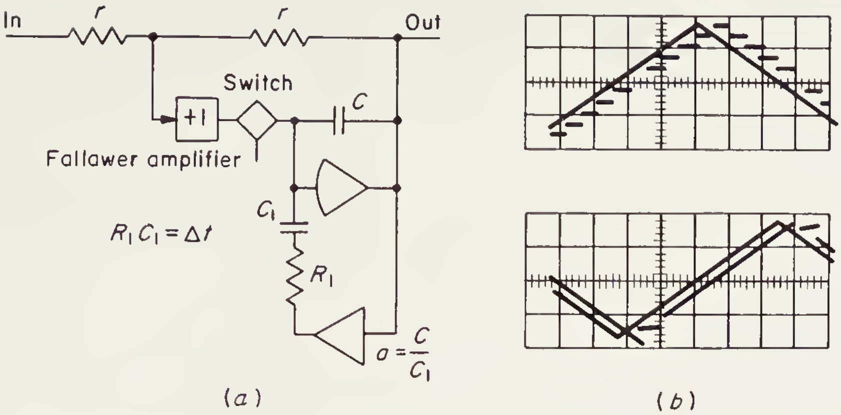


Fig. 9-3. Simple modification of a zero-order-hold circuit for first-order-hold operation. The a-c-coupled d-c amplifier need not be a low-drift amplifier, so that this circuit is a very economical choice for data reconstruction following DACs, analog multiplexers, sampled-data time-delay circuits, etc. Figure 9-3*b* shows zero-order-hold and first-order-hold input and output traces with $C = C_1 = 0.05\ \mu$F, $R = 20$K, $\Delta t = 1$ msec; the input trace was delayed slightly in order to show the waveform more clearly. This circuit can be driven by a DAC without double buffering; the DAC can be loaded while the sampling switch is OFF.

into C_1 be equal to $-C\,\Delta Y_0/\Delta\tau = -C\,\Delta Y_0/\Delta t$, where Δt is the time between period samples. It follows that we must have $R_1C_1 = \Delta t$.

The two-DAC circuits of Fig. 9-4 produce outputs of the form

$$Y = Y_k + a_k(t - t_k)$$

where a_k can be chosen to implement extrapolation or, with suitable prediction (Sec. 9-5), interpolation.

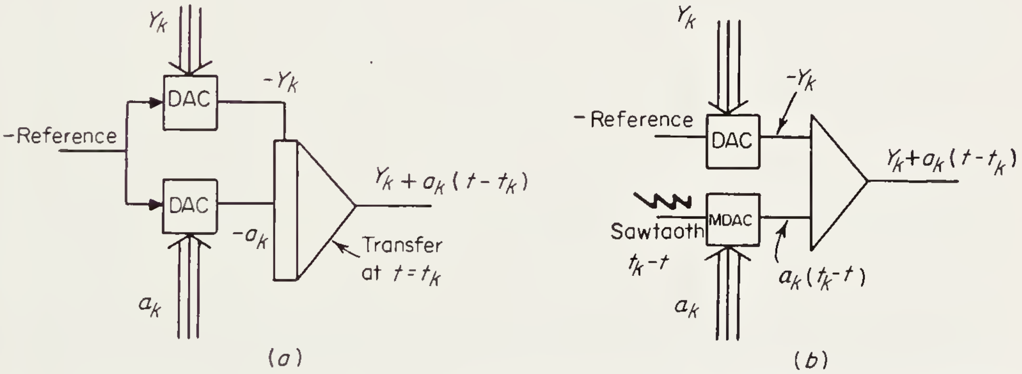
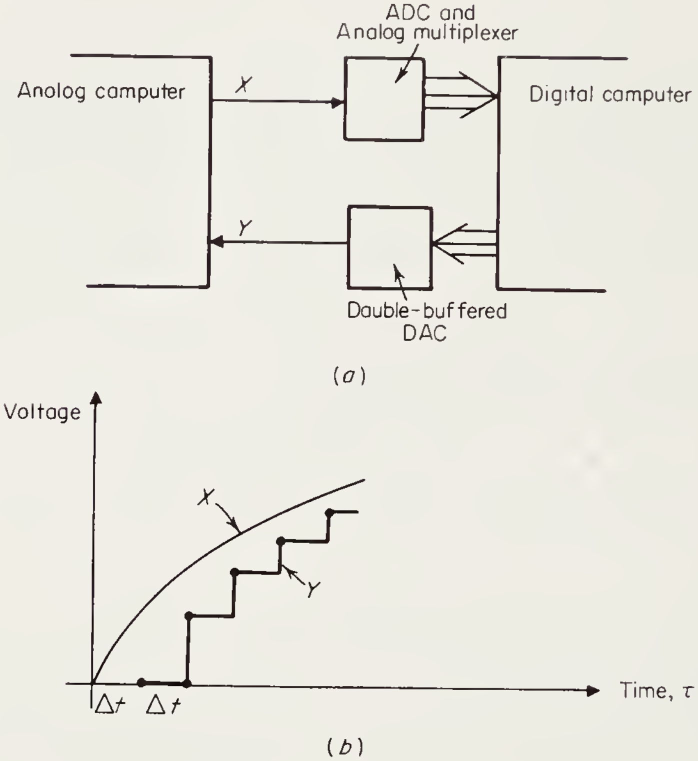


Fig. 9-4. Two-DAC circuits for extrapolation or (predictive) interpolation. If the finite integrator or sawtooth resetting time is objectionable, the analog-computer state-variable integrators are placed into HOLD during this time.

9-4. Errors Due to Processing Delays in Combined Analog-Digital Simulation. In *closed-loop combined simulation* (Figs. 1-20 and 9-5*a*), analog signals are fed back to the analog computer following successive A/D conversion, digital processing, and D/A conversion. The finite time required for these operations (some of which are time-shared between multiple variables) determines the minimum usable sampling period or *frame time* Δt. Errors due to the finite processing time are added to the data-reconstruction errors in the manner illustrated in Fig. 9-5*b*.

The serious nature of time-delay errors was already discussed in Sec. 3-2. Referring to Fig. 9-5*b*, we see that a sinusoidal signal $Y = a \sin 2\pi f t$ is subject to a maximum percentage error

$$\epsilon = 400\pi f \, \Delta t \tag{9-1}$$



(*a*)

(*b*)

Fig. 9-5. Combined analog-digital-simulation loop (*a*), and effect of processing delay and zero-order hold on a sine wave (*b*). Frame time Δt and errors are exaggerated.

For a processing delay less than 25 msec (which is fairly typical) and 100 samples per sine wave, Eq. (1) reveals a maximum error of over 12 percent at a sine-wave frequency as low as 0.4 Hz. It does not require very complicated error analysis to see that such errors, if left uncompensated, could spell the doom of combined analog-digital simulation.

The situation may be even worse, for combined simulation will involve closed-loop feedback. The sine-wave delay error expressed by Eq. (1) corresponds roughly to a spurious phase lag of $4\pi f \, \Delta t$ radians introduced into the feedback loop of Fig. 9-5*a*. This amounts to over 7 deg in our simple example. Such phase errors, if left uncompensated, will affect the loop stability; depending on the specific hybrid-computer loop, they can amplify other errors or even cause completely unstable behavior.

The sampled-data errors in a general nonlinear combined-simulation system are not amenable to simple analysis but are best studied by repeated computations with different time scales and/or frame times. If we linearize the nonlinear-system equations of our analog-digital sampled-data system in the manner of Sec. 8-2, we shall obtain the performance equation of a linear sampled-data system which can be regarded as roughly time-invariant at each instant of time. The stability and possible compensation of this system can then be studied by the z-transform methods familiar to control engineers. This has been done in Refs. 6 to 15 for a number of special cases and applications. Such approximate analysis can be extended to different data-reconstruction and compensation methods.

9-5. Reducing Sampled-data Errors. (a) General Remarks. As we have seen, meaningful closed-loop combined analog/digital simulation will require some method for compensating the sampled-data errors due to processing delay and data reconstruction. Even the best data-reconstruction filter (Sec. 9-3) can reduce the combined error (1) at best by one-half. The following compensation methods can alleviate the situation by at least a factor of 10 but cannot be applied blindly, since each compensation technique may introduce errors of its own.

(b) Intermittent HOLD. One way to compensate for processing-delay errors is to place the analog computer into HOLD as soon as all analog voltages have been sampled and to return to the COMPUTE mode only after all DACs or MDACs have been updated (this technique, incidentally, does away with the need for double-buffered DACs). Since the analog computer is now stopped during the conversion and digital-processing delay, the delay error (but not the data-reconstruction error) is effectively eliminated. Digital computation ceases during the analog-computing periods, whose length Δt is then determined only by frequency-response considerations, not by digital processing time.

The repeated switching operations will cause accumulation of switching-spike charges on the integrator capacitors, but the resulting offset is not too serious in *slow* analog computation with 1-μF integrating capacitors: 2,000 1-volt pulses applied to a 1-μF capacitor through a 10-pF switch capacitance add up to only 0.02 volt (Sec. 5-1). Differential integrator switching times might also cause cumulative timing errors. In simulations involving partial-system tests of real equipment, intermittent-HOLD waveforms may not be sufficiently realistic, although it is possible to speed up the time scale to make up for the intermittent-HOLD action.

(c) Digital-computer Prediction. The digital computer can readily substitute *predicted values* like

$$Y'_k = Y_k + \frac{T_P}{\Delta t}(Y_k - Y_{k-1}) \tag{9-2}$$

for each DAC output Y_k, and thus approximately compensate for the processing delay and improve data reconstruction (we can predict $T_P = \frac{3}{2}\Delta t$ ahead for zero-order reconstruction and use $T_P = 2\Delta t$ for a predicted interpolative hold). Higher-order prediction polynomials may be substituted for Eq. (2) to improve prediction; it is also possible to combine prediction with integration routines employed in the course of digital processing.[11]

Digital prediction is attractive especially with interpolative-hold circuits, but digital prediction involves a subtle danger. When we introduce difference-equation prediction formulas like Eq. (2) or higher-order prediction formulas, *we are raising the order* of the differential/difference-equation system representing our closed-loop model. It has been shown that this can introduce spurious and possibly unstable high-frequency modes. The resulting errors depend on the system being simulated; Refs. 8 to 15 discuss several special cases.

(d) Analog-computer Prediction.[4] We can also introduce predicted data *ahead* of the analog-to-digital conversion in Fig. 9-5*a*. Figure 9-6*a* shows how an ADC input

$$X'(\tau) = X(\tau) + T_P\frac{dX}{d\tau} \approx X(\tau + T_P) \tag{9-3}$$

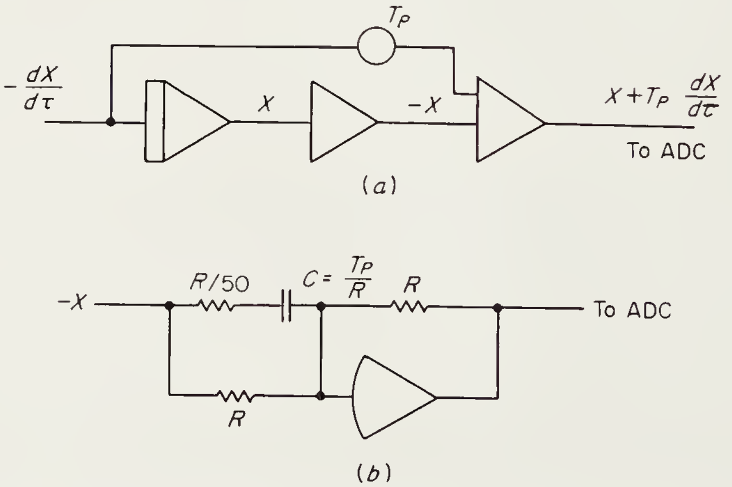


Fig. 9-6. Analog prediction circuits for processing-delay compensation (see also Fig. 1-28).

(with $T_P + \frac{3}{2}\Delta\tau$ for zero-order-hold reconstruction) can be formed if a voltage proportional to the time derivative $dX/d\tau$ is available in our analog-computer setup. If this is not true, the differentiating circuit of Fig. 9-6*b* still permits approximate analog prediction; but note that this scheme again increases the order of our differential-equation system by one and may lead to spurious modes.

The analog predictors of Fig. 9-6 may have switchable capacitors to accommodate different frame times Δt; but they will work only for *constant* frame time.

(e) Perturbation and Hybrid-code Methods. As we have seen in Secs. 2-4, 5-9, and 8-2, computing-error effects can be reduced dramatically if they affect only *small perturbations* of the computer variables.

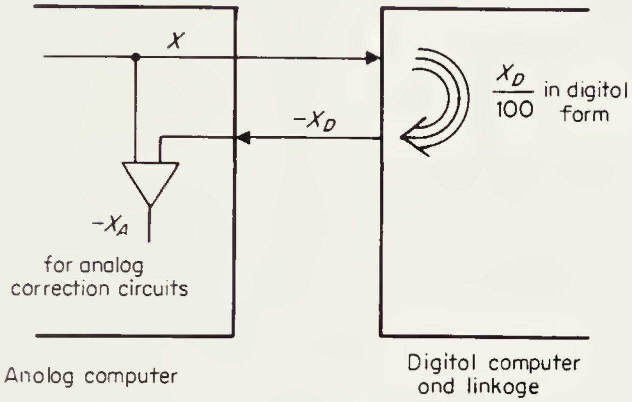


Fig. 9-7. Hybrid-code technique.

Where such perturbation techniques apply, they are without question the best way to deal with hybrid-computer errors. The application of perturbation methods is an art rather than a science. In particular, the digital computer has been used to add complicated but relatively small nonlinear corrections to analog voltages computed with accurate amplifiers, coefficient-setting units, and multipliers.[59]

In the computer setup of Fig. 9-7, the digital ADC output is reconverted into an analog voltage X_D and subtracted from the ADC input X to produce an error voltage

$$X_A = X - X_D \tag{9-4}$$

which includes effects related to zero-order-hold operation, quantization, and ADC errors ("analog-to-hybrid-code" conversion; see also Sec. 9-8*c*). In suitable applications, such error voltages can be processed in the *analog computer* to produce suitable corrections to the DAC outputs Y. The accurate hybrid multipliers of Fig. 6-20 and the hybrid function generators of Figs. 9-9 to 9-14 are examples of this technique; note that processing delays are compensated in Fig. 9-7. Reference 5 discusses additional correction methods.

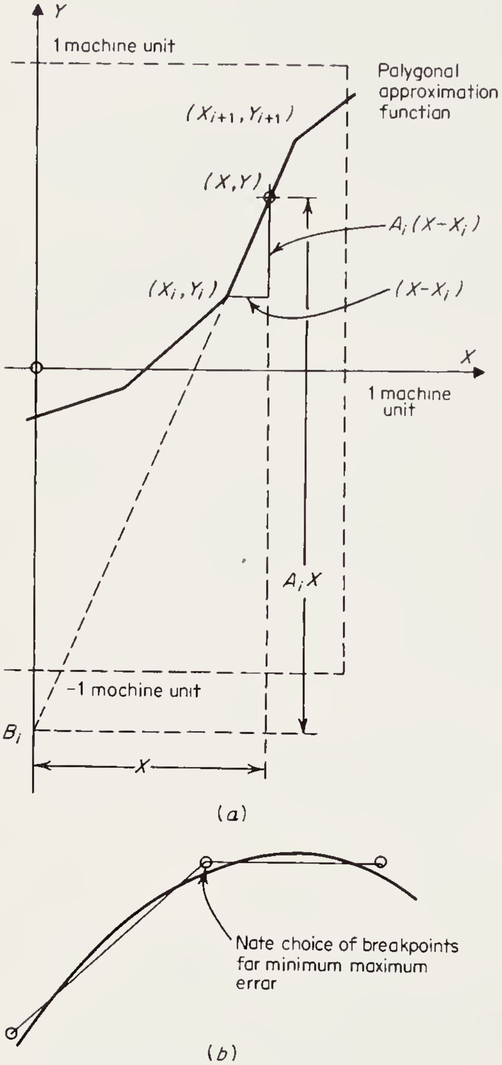


Fig. 9-8. Approximation of a function $Y = Y(X)$ by linear interpolation between tabulated points (*a*) and function fitting (*b*).

9-6. Conclusions. Closed-loop combined analog-digital simulation involves possibly serious sampled-data errors. To realize the advantages of analog-computer speed and digital accuracy and flexibility in combined simulation, we must be sure sampled-data errors can be kept tolerable by:

1. Models involving only sufficiently low frequencies in the digital part of the loop
2. Restriction to relatively fast digital operations (e.g., direct-memory-access time-delay simulation, function generation, Secs. 9-7 to 9-9)
3. Perturbation methods

Combined simulation involves expensive hardware and software as well as knowledgeable personnel for programming and checkout. Such expenses are justified if combined simulation has a significant speed advantage over all-digital simulation because:

1. There is no other practical way to obtain the requisite bandwidth, e.g., for partial system tests, and/or
2. The simulation study is so large (in complexity and/or number of runs) that digital simulation would be more expensive.

By contrast, hybrid computation involving only digital supervision of analog-computer runs (parameter, optimization, and statistical studies,

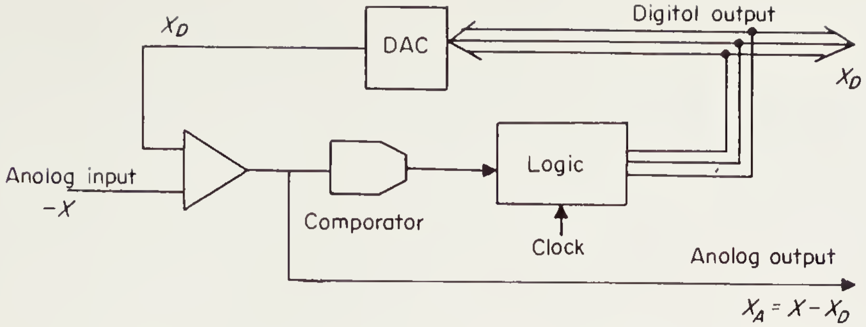


Fig. 9-9. Analog-to-hybrid-code converter. The logic shown can implement either "continuous" or successive-approximation conversion (Sec. 5-13). Note that the DAC used can have low *resolution*, but it must be as *accurate* as the required accuracy of X.

some partial-differential-equation studies) tends to combine analog and digital techniques much more advantageously, is easier to program, and works nicely with small digital computers.

HYBRID-COMPUTER FUNCTION STORAGE AND FUNCTION GENERATION

9-7. Function Storage. Fixed and Variable Time Delays. In general-purpose hybrid computers, *core-storage implementation of function storage or time delays* has superseded all other methods (see also Secs. 2-19 to 2-22, 8-16*c*, 8-21, and 8-23). To generate

$$Y(\tau) = X(\tau - \tau_D)$$

where τ_D is a *fixed* time delay, one stores successive samples $X(k\ \Delta t)$ in a block of memory locations; these samples are read out $m\ \Delta t = \tau_D$ sec later. The number of storage locations reserved is the number of Δt intervals needed to make up τ_D or, if a variable is to be stored for another computer run, the number of Δt intervals in one computer run. When either the read-or-write-address pointer reaches the last memory location of the block, these pointers are reinitialized to start at the beginning, so that "recirculating" storage is obtained.

Such a recirculating-storage scheme can be implemented with *programmed instructions* synchronized to clock interrupts,[20] but a *direct memory-access technique* saves programming overhead and permits much greater computing speed at the expense of some data-channel hardware.[21] The DMA method of time-delay simulation utilizes two data channels like that shown in Fig. 7-22; the two current-address counters serve as hardware pointers to the read and write locations. Periodic data-channel read and write requests are obtained from the analog-computer clock. When either data-channel word counter overflows to indicate that the end of the storage block has been reached, a short high-priority interrupt-service routine reinitializes the corresponding current-address counter and the word counter (Sec. 7-12).

Since many DMA channels can handle over 10^6 samples/sec, the maximum sampling rate will be determined by the ADC conversion rate available; a 100-KHz sampling rate is readily possible if an ADC is committed to a function-storage channel. This permits one to delay signals containing frequency components up to 5 KHz, especially with first-order-hold data reconstruction (Sec. 9-3). Function-storage channels can steal memory cycles from a suitable general-purpose computer, or a separate minicomputer or special-purpose memory with its own data-channel counters can serve several channels. Current-address and word counters implemented in minicomputer memory registers minimize the interface hardware required at some expense in computing speed.[21]

The recirculating core-storage technique can also implement *variable transport delays.* We now record samples $X(k\,\Delta t)$ with a fixed-period clock as before, while the delayed samples are read out at a *variable* clock rate proportional to the derivative $d\tau_D/d\tau$ of the variable time delay $\tau_D(\tau)$. Assuming that an analog voltage proportional to $d\tau_D/d\tau$ (e.g., flow velocity in a pipe causing the transport delay) is available in the analog computer, we can readily generate variable-frequency interrupt or DMA-request pulses with the accurate frequency-modulator circuits of Fig. 4-28.[19,21]

With fixed time delays, read and write pulses can be spaced to avoid mutual interference, but variable time-delay data-channel requests require the usual priority circuit (Fig. 7-19), since read and write requests might overlap. It is probably preferable to assign priority to the write operation.

9-8. Hybrid-computer Techniques for Function Generation. (a) Introduction. Setting even a single-variable diode function generator is a task most analog-computer programmers would rather avoid (Sec. 4-15), although the better types of card- or patchboard-programmed DFGs have improved this situation somewhat. Hybrid-computer function generation is more convenient as well as more accurate and is almost mandatory for general-purpose generation of functions of two or more variables. While either analog or digital computers can implement *polynomial function approximations*, this section will be concerned with general-purpose table-lookup techniques. It is clear that a digital computer can readily store and retrieve *tables* of function values or of function values and argument values. Simple table lookup, however, is rarely accurate enough, and *interpolation* (usually linear interpolation) is required. For functions of one variable, we can approximate a given piecewise-continuous function

$$Y = Y(X) \tag{9-5}$$

by a piecewise-linear (polygonal) approximation function

$$F(X) = A_iX + B_i \qquad (X_i \leq X < X_{i+1}) \tag{9-6}$$

or

$$F(X) = A_i(X - X_i) + Y_i \qquad (X_i \leq X < X_{i+1}) \tag{9-7a}$$

where

$$A_i = \frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \qquad B_i = Y_i - A_iX_i \tag{9-7b}$$

(Fig. 9-8*a*), with suitably fitted points X_i, Y_i (Fig. 9-8*b*). Note that the X, Y, X_i, and Y_i are scaled between -1 and 1 machine unit, while this is not necessarily true for the A_i and B_i. References 60 to 62 discuss optimal polygonal-function approximations.

Since both arguments and function values will appear as analog variables, they must be scaled in any case, and we shall save core storage and digital processing time through *fixed-point* data representation in the digital computer. Most hybrid-computer methods employ *nonuniform breakpoint spacing* to minimize the number of breakpoints needed for a given accuracy, but *uniform breakpoint spacing* eliminates the need to store breakpoint abscissas.

(b) Digital Table Lookup and Digital Interpolation. Digital-computer table lookup becomes especially simple if we employ $2^N + 1$ breakpoints uniformly spaced between $X_0 = -1$ and $X_{2^N+1} = +1$, for then the *n*-bit two's-complement code for

$$X = X_i + (X - X_i)$$

has the form

$$\underbrace{\boxed{\text{Sign bit}}\boxed{N - 1 \text{ bits}}}_{\text{represents } X_i}\underbrace{\boxed{n - N \text{ bits}}}_{\text{represents } X - X_i}$$

The N left-hand (most significant) bits represent the desired breakpoint abscissa X_i *and can be used directly to find a memory address pointing to the corresponding function value* Y_i. The remaining $n - N$ bits represent the difference $X - X_i$ needed for digital interpolation.[35] This method produces the expression (7*a*) in digital form within 20 to 50 μsec with fast digital computers. Reference 35 also describes a simple extension of the method to nonuniform breakpoint spacing.

All-digital interpolation is not unattractive but leaves us with the usual data-reconstruction errors (Sec. 9-3). *Analog interpolation* methods generate a portion of the interpolation polynomial (6) or (7) directly in the analog computer; this reduces data-reconstruction errors as well as the amount of digital computation.

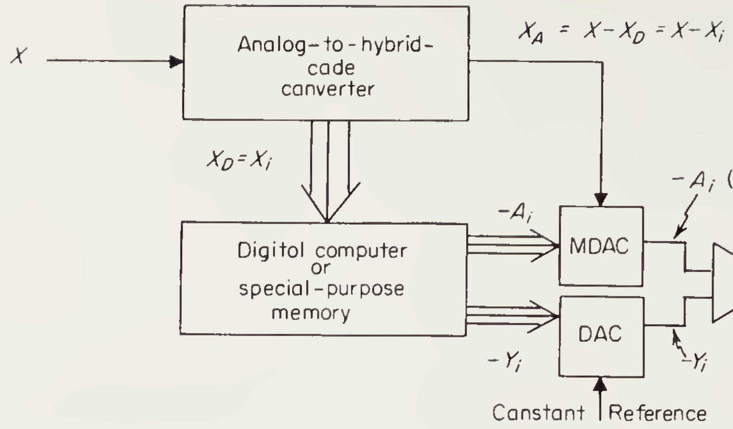

X
Analog-to-hybrid-code converter
$X_A = X - X_D = X - X_i$
$X_D = X_i$
Digitol computer or special-purpose memory
$-A_i$
MDAC
$-A_i(X-X_i)$
Y
DAC
$-Y_i$
$-Y_i$
Constant Reference

Fig. 9-10. Hybrid-code-interpolation function generator using $2^N + 1$ *uniformly spaced* breakpoints. Note the simplicity of this scheme; a read-only MOSFET memory can be used to generate special functions.

(c) Digital Table Lookup and Hybrid-code Interpolation Using Uniformly Spaced Breakpoints. Feedback-type A/D converters (either "continuous" or successive-approximation converters, Sec. 5-13) with analog input X are readily designed to output the explicit analog quantization error $X_A = X - X_D$ as well as the desired digital output X_D (Fig. 9-9). The converter thus produces a *hybrid-code representation* of the analog input X in terms of X_D and X_A:

$$X = X_A + X_D \tag{9-8}$$

Such a hybrid-code representation was already used in Fig. 6-20 to produce accurate and fast multiplication; Refs. 25 to 31 describe other

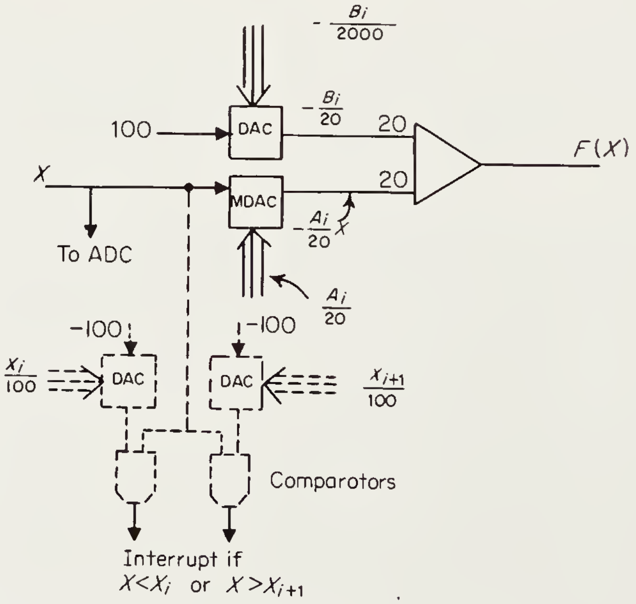


Fig. 9-11. This simple hybrid function generator implements $Y \simeq F(X) = A_i X + B_i$ $(X_i \leq X < X_{i+1})$. Breakpoint-crossing comparators are optional.

applications. Figure 9-10 shows a hybrid-code-interpolation function generator whose remarkable simplicity is predicated on the use of $2^N + 1$ *uniformly spaced* breakpoints X_i, so that $X_D = X_i$.

(d) Digital Table Lookup and Analog Interpolation. The hybrid function-generator schemes of Figs. 9-11 and 9-12 permit curve fitting with nonuniform breakpoint spacing. The digital computer stores a table of breakpoints X_i as well as the function values Y_i and/or the coefficients A_i, B_i (the latter can be computed or precomputed and stored). The optional comparators shown make it possible to perform interrupt-driven conversion and table-lookup operations only whenever the argument voltage X crosses a breakpoint level X_i.

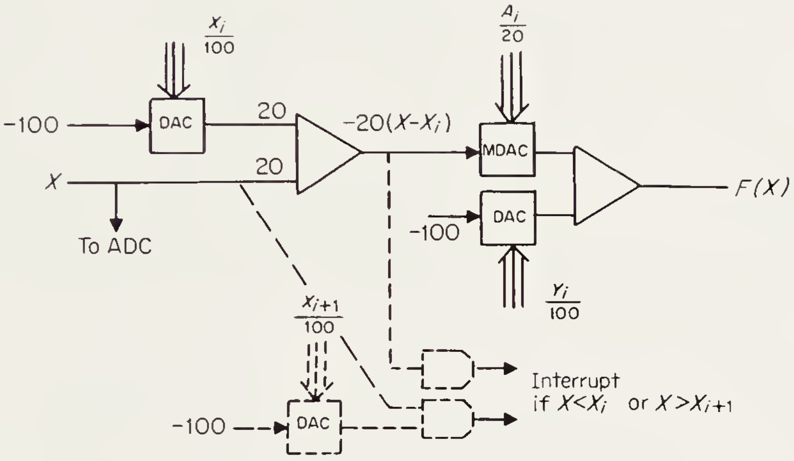


Fig. 9-12. Hybrid function generator for

$$Y \simeq F(X) = A_i(X - X_i) + Y_i \qquad (X_i \leq X < X_{i+1})$$

Figure 9-13, modified from a design by A. I. Rubin,[34] implements linear interpolation in the form

$$Y \approx F(X) = Y_i\left(1 - \frac{\Delta X}{\Delta X_i}\right) + Y_{i+1}\frac{\Delta X}{\Delta X_i}$$

with $$\Delta X = X - X_i \qquad \Delta X_i = X_{i+1} - X_i$$

$$(X_i \leq X < X_{i+1}) \tag{9-9}$$

The extra cost of generating the voltages $100\Delta X/\Delta X_i$ and $100(1 - \Delta X/\Delta X_i)$ will be justified when we generalize the circuit to the case of two or more variables, where it will be used several times (Sec. 9-9).

9-9. Functions of Two or More Variables. The table-lookup/interpolation techniques of Sec. 9-8 have been extended to functions of two, three, and four variables with up to 20 breakpoints in at least one variable; 5 to 8 breakpoints in each variable are more typical.

Implementation of multi-variable interpolation formulas like

$$F(X, Y) \approx A_{ik}X + B_{ik} + (C_{ik}X + D_{ik})Y$$

$$(X_i \leq X < X_{i+1},\ Y_k \leq Y < Y_{k+1}) \tag{9-10}$$

with summers, MDACs, and analog multipliers (Fig. 9-14) requires the least hardware, but we must compute (or precompute and store) all the coefficients A_{ik}, B_{ik}, C_{ik}, D_{ik} for the given function values $F_{ik} = F(X_i, Y_k)$. In addition, scaling may be troublesome (see also Sec. 9-10).

Rubin[34] suggests two-variable interpolation through a combination of three one-variable interpolations:

$$\left.\begin{aligned}
F(X_i, Y) &= F_{ik} + \frac{\Delta Y}{\Delta Y_k}(F_{i,k+1} - F_{ik}) \\
F(X_{i+1}, Y) &= F_{i+1,k} + \frac{\Delta Y}{\Delta Y_k}(F_{i+1,k+1} - F_{i+1,k}) \\
F(X, Y) &= F(X_i, Y) + \frac{\Delta X}{\Delta X_i}[F(X_{i+1}, Y) - F(X_i, Y)] \\
&= F_{ik}(1 - \delta X - \delta Y + \delta X\, \delta Y) \\
&\quad + F_{1+1,k}(\delta X - \delta X\, \delta Y) \\
&\quad + F_{i,k+1}(\delta Y - \delta X\, \delta Y) + F_{i+1,k+1}\, \delta X\, \delta Y
\end{aligned}\right\} \tag{9-11}$$

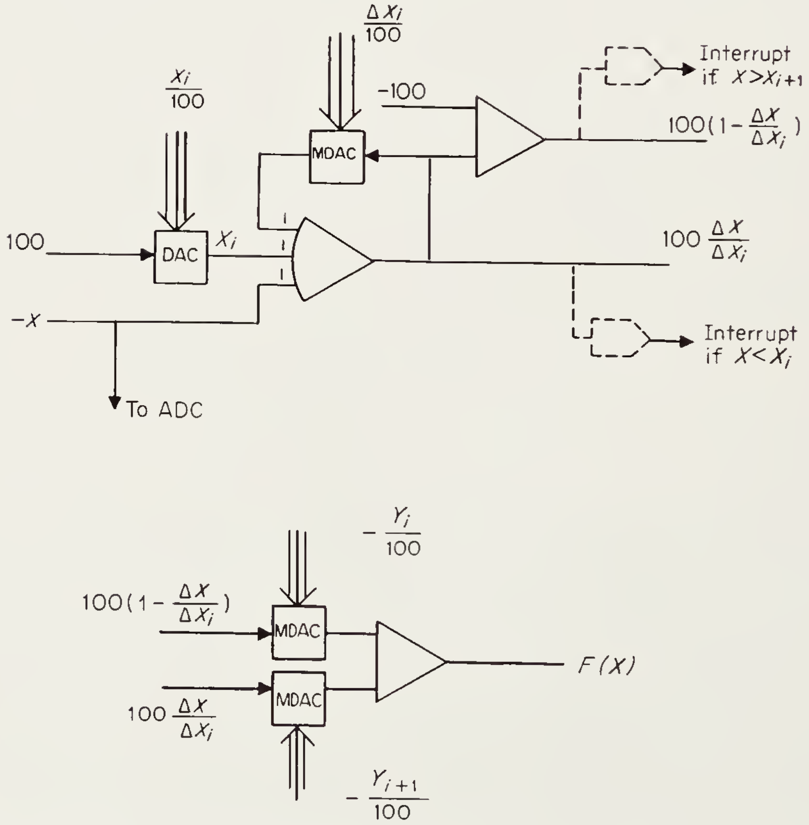


Fig. 9-13. "Modules" for generating voltages proportional to $\Delta X/\Delta X_i$, $1 - \Delta X/\Delta X_i$, and $F(X)$ according to Eq. (9-9).

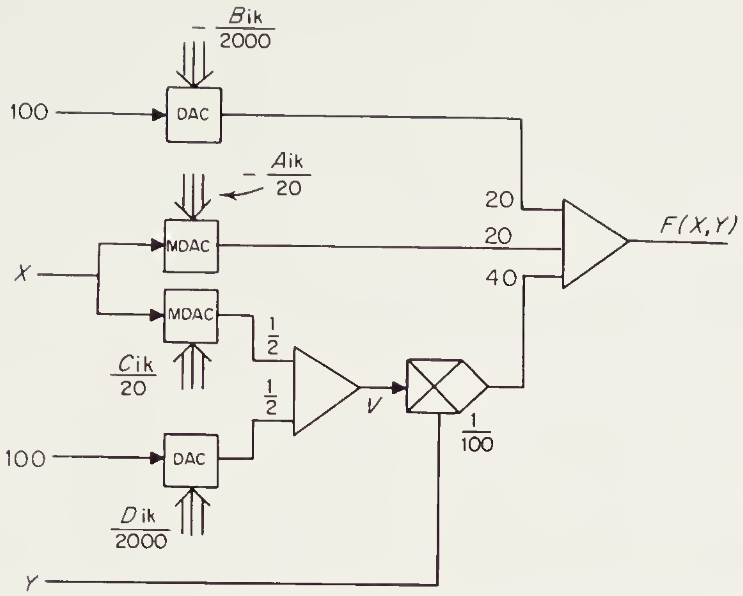


Fig. 9-14. Principle of a two-variable function generator implementing Eq. (9-10). This is not a practical general-purpose circuit, because good accuracy would require us to *rescale* V for different coefficient combinations (see also Sec. 9-10).

where

$$\delta X = \frac{\Delta X}{\Delta X_i} = \frac{X - X_i}{X_{i+1} - X_i} \qquad \delta Y = \frac{\Delta Y}{\Delta Y_k} = \frac{Y - Y_k}{Y_{k+1} - Y_k}$$

$$(X_i \leq X < X_{i+1},\ Y_k \leq Y < Y_{k+1})$$

Voltages proportional to $\delta X = \Delta X/\Delta X_i$ and $\delta Y = \Delta Y/\Delta Y_k$ are generated as in Fig. 9-13; $F(X,Y)$ is then readily obtained with four MDACs, an analog multiplier, and some summing amplifiers. The method can be generalized for three and four variables.

9-10. Discussion. The hybrid function-generator schemes of Figs. 9-11 to 9-14 require only standard analog computing elements and converters, an advantage both in special-purpose and general-purpose computers. Each method becomes more efficient when we want more than one function of the same variable or variables, provided that we can use the same breakpoint arguments for these functions. This is especially true for the circuit of Fig. 9-13 and for multivariable function generators derived from it.

Assuming that it is reasonable to allow for function slopes as large as ±20 volts/volt, Figs. 9-11 and 9-12 each include a gain-of-20 amplifier. Such high gain tends to compromise scaling and bandwidth even for argument values where the large slope is not needed. The circuit of Fig. 9-13 does *not* have this drawback *if we agree to provide small breakpoint intervals* ΔX_i *wherever the slope is large;* variable breakpoint spacing is an advantage in this connection.

It is feasible to rescale Fig. 9-11 by implementing

$$F(X) = C_i(A_iX + B_i) \qquad (X_i \leq X < X_{i+1})$$

but we must now store and set C_i as well as A_i and B_i for each breakpoint interval. C_i might be restricted to the values 1 and 16, so that a 1-bit MDAC (i.e., a simple analog switch) could change the summing-amplifier feedback resistor. Figure 9-12 could be similarly rescaled, but in each case the basic simplicity is lost.

Uniform breakpoint spacing is not always practical, especially with functions of two or more variables (where the numbers of breakpoints used for each argument *multiply*). Uniform breakpoint spacing simplifies Figs. 9-12 and 9-13. If it is possible to fit $2^N + 1$ uniformly spaced breakpoints, digital table lookup is greatly simplified (Sec. 9-8*b*) and can, in fact, be implemented through direct-memory-access addressing without programmed instructions (Sec. 7-12). This technique would permit very fast function generation.

PROGRAMS AND SOFTWARE FOR HYBRID ANALOG-DIGITAL COMPUTATION

9-11. Software Requirements. We will require computer programs which, through appropriate digital and linkage operations, operate the hybrid-computer system to:

1. *Set up* initial conditions, coefficients, functions, and switching connections, either *individually* on operator command or *in blocks.*
2. *Read* selected voltages in the analog computer.
3. *Static-check* analog-computer setups (Sec. 7-6*c*). This requires precomputation of static-check values as well as readout and printout.
4. *Execute* differential-equation-solving computer runs and multirun programs (Sec. 9-13).
5. *Collect and process data* from these computations for display and report preparation.

These hybrid-computer programs may require an extended (or even new) *source language* or, at least, a set of hybrid-computer *library routines* added to an existing source language such as FORTRAN. To program and run problems, we shall require *system programs* which permit us to:

6. *Create and modify programs* in a convenient source language. This implies an *editor* program (or off-line editing), a *language processor* (compiler, assembler, interpreter, and/or precompiler/translator), plus a *linking loader* for loading object-language and library programs.
7. *Call system programs for all the above functions* into core, as required (usually from a disk), and also to *store and recover each user's own files* containing programs, data, and results (usually from tape).

The "executive" program for the latter purpose is called a *hybrid executive* or hybrid monitor and will also abort programs and display diagnostic messages when programming errors or equipment malfunctions occur. The executive may also incorporate facilities for *time sharing* the digital computer and for allocating linkage and other input/output hardware among multiple users.

An ideal hybrid-computer program package would accept a problem statement specifying the problem differential equations, compute some digital check solutions, extract appropriate analog scale factors, and compile an analog-computer setup for automatic patching (Sec. 7-4*c*). The program would next precompute and execute the static check (Sec. 7-6*c*), run the problem, process data, and prepare a report. Such software, while entirely feasible, has only been approximated.

9-12. Library Routines and Their Use in Programs and with Interpreters. For programming convenience (and brevity), hybrid-computer programs will need a library of routines for hybrid-computer operations, such as:

Select specified analog component for readout
Read specified ADC or digital voltmeter
Set specified coefficient unit or potentiometer
Load specified MDAC buffer register
Update specified MDAC
Load logic-control register with specified binary number
Start analog-computer run
Read specified sense line
Read data from (or set data into) analog components N to M

In addition to these basic linkage-operation routines, we need utility programs for time-delay simulation (Sec. 9-7), function generation (Secs. 9-8 to 9-10), integration, prediction (Sec. 9-5*d*), etc., plus the usual mathematics library.

Library routines for a given type of hybrid computer are usually programmed and assembled once and for all, and are filed on a binary tape. They are generally written by an expert programmer, for a deceptively simple input/output operation can involve fairly complicated details:

Start a device (e.g., ADC)
Check for proper operation
Wait for interrupt (e.g., conversion complete, potentiometer set)
On interrupt, save (and later restore) active registers, modify priorities
Transfer data
Perform parity check
Change data format (fixed-point to floating-point, etc.)
Restore device operation (reset flags) etc.

Hybrid-computer library routines will be used in two different ways:

1. They can be loaded with assembly-language, FORTRAN, ALGOL, or simulation-language programs, which call hybrid-computer operations as subroutines.
2. They can be used "manually" (i.e., in response to commands from a console typewriter or cathode-ray-tube/keyboard) for checkout and between computer runs to read voltages, set coefficients and logic states, etc.

"Manual" operation does not involve execution of complete assembled or compiled programs but requires an *interpreter* which immediately translates individual commands into calls for library routines. Special interpreters have been written for this relatively simple job, but adaptations of existing interpreter systems such as BASIC or FOCAL will do nicely.

Interpreter control of the hybrid computer is preferable to manual operation of analog-computer keyboards, switches, etc., between runs, because the digital computer can keep a complete record of all changes made in large computer setups. Analog-computer controls are best used for maintenance purposes only.

9-13. Typical "Main Programs." (a) Data Processing. Analog data are sampled and converted in response to clock pulses or other events (comparator responses, logic, alarm sensors). There are two different modes of operation:

1. If the required digital operation can work with partial data (e.g., accumulation of averages), *and* if the data rate is sufficiently slow, we use conversion-completed *interrupts* to transfer data to the digital computer, to begin each partial operation, and to disable program switches inhibiting digital operations with unavailable data.
2. If data rates are fast, or if a block of data is needed before digital computation can begin, we use direct-memory-access block transfer of data (data channel, Sec. 7-12). We send an interrupt when the desired block is finished.

(b) Digitally Controlled/Iterative Analog Computations. Digital, analog, and linkage operations for this type of hybrid computation, which was discussed in Secs. 1-13 and 8-13 to 8-16, are listed in their proper time sequence in Table 9-2. Referring to Sec. 7-3, we note that *sampling times* for various analog variables, and also *the end of each analog-computer run*, will be determined by digitally preset counters counting analog-computer clock pulses, by analog-comparator responses,

Table 9-2. Digitally Controlled/Iterative Analog Computation

Digital computer	Linkage	Analog computer (includes timing and logic circuits)
Determine initial parameters		INITIAL RESET
START LINKAGE OPERATION →	Set coefficients, MDACs, logic	
	START ANALOG RUN →	COMPUTE Determine sampling times and end of run
(Idle)	Read ADCs, comparators, logic Transfer data to digital computer	← RESET
Compute new parameters Process and store data for report	← INTERRUPT DIGITAL COMPUTER Reset linkage logic	
Test for end of study		
START LINKAGE OPERATION →	(Repeat operations as above until study is terminated)	

and/or by patched logic. It is, for instance, possible to terminate an analog computer run as soon as all sampling pulses have been triggered. Sampling pulses place sample-hold circuits or integrators into HOLD at the correct sampling times (and can also start analog-to-digital conversion if individual rather than multiplexed ADCs are used).

(c) Closed-loop Combined Analog-Digital Simulation. As we have seen (Sec. 9-5), this is usually a time-critical procedure. Unless we employ intermittent-HOLD operation (Sec. 9-5*b*), operations flow as outlined by Table 9-3.

(d) Discussion. DMA Operations and Hardware/Software Tradeoffs. Note that the digital-computer operations listed in Tables 9-2 and 9-3 are all interrupt-driven, so that a time-shared digital computer can return to a background program instead of idling during analog-computer operation. Very small digital computers can, instead, idle

Table 9-3. Closed-loop Combined Analog-Digital Simulation

Digital computer	Linkage	Analog computer (includes timing and logic circuits)
Determine initial MDAC settings SET UP LINKAGE AND FRAME TIMING START LINKAGE OPERATION →	Load MDAC buffers Update all MDACs Start analog run and frame timer →	RESET COMPUTE
Compute new MDAC settings (includes prediction) Test for end of run RESET LINKAGE WHEN FINISHED →	Load MDAC buffers Update all MDACs ← Read ADC channels ↙ INTERRUPT DIGITAL COMPUTER	Count clock pulses for *one frame time* Δt, then TRIGGER SAMPLE HOLDS AND LINKAGE
Next frame starts: frames repeat until stopped by operator, timing/logic circuits, or digital program		

and sense the end of an analog-computer run (Table 9-2) or of the current combined-simulation frame (Table 9-3) with a sense/skip loop (Sec. 7-10).

Unless there are very few ADC channels, MDACs, and coefficient units, the linkage operations indicated in the center columns of Tables 9-2 and 9-3 ought to employ cycle-stealing data channels for data transfer directly to and from memory (Sec. 7-12). This will save time, memory, and programming effort. *With direct-memory-access linkage operations, the digital-computer programmer simply refers to analog input and output data as to any other data stored in memory.* Much linkage programming is replaced by hard-wired data-channel logic, which ties complete linkage-operations sequences (center columns of Tables 9-2 and 9-3) into the analog-computer clock/logic system.

An analog-computer *overload* causes a logic response in the analog-computer digital patchbay. Patched logic then places the analog computer into RESET or HOLD and also interrupts the digital computer; the overload-interrupt service routine will cause the digital computer to disregard data from this run, change parameters so as to alleviate the overload condition, or stop the study.

(e) Interrupt Servicing and Reentrancy. Apart from the many interrupt-servicing routines required by linkage operations, there may be program interrupts *directly required by the nature of a hybrid-computer simulation.* Such situations arise from various limit or alarm conditions in simulated process-control systems, or when a simulated aerospace vehicle drops a rocket stage or lowers flaps or landing gear. If such a condition happens to be detected in the analog computer (say by a comparator indication), it will be necessary to interrupt the digital-computer program for suitable modifications; a portion of the old program, e.g., trajectory computation, will have to be *resumed* after the interrupt is serviced. Such general-purpose interrupt-service programs may require the same library subroutines both in the main program and the interrupt-service routine. If a library routine requiring temporary storage is interrupted and then called again by the interrupt routine, data in storage may be overwritten and destroyed, so that the interrupted program cannot be properly resumed. *Reentrant programming* avoids this sort of trouble. A primitive reentrancy technique is to provide duplicate copies of library subroutines, but this is cumbersome and does not help when the same routine is quickly called more than twice. Another primitive method is to write library subroutines with such high software priorities that external interrupts are honored only after these routines are completed; but this slows interrupt servicing badly. Proper reentrant programming of subroutines places all temporary storage into a stack addressed with the aid of a stack pointer, which is incremented by new interrupts and decremented on completion of a subroutine. FORTRAN compilers which make all subroutines (not only some library routines) reentrant are sometimes said to generate *real-time* FORTRAN. Such compilers are very suitable for hybrid computation and also permit programs with recursive subroutines.

9-14. Operating Systems. The best way to control small as well as large hybrid-computer systems is through a *CRT-keyboard terminal*, with a line printer or automatic typewriter near the operator's position producing hard copy only when it is really wanted. CRT-keyboard terminals plug into computer terminations originally designed for teletypewriters but are much faster, more reliable, and more convenient.

A mass storage device (disk or drum) is also almost a necessity for smooth operation and permits the use of a *hybrid-monitor* or *hybrid-executive program* for rapid loading and filing of users' programs as well as of service programs used for static checking, editing, compiling, assembling, and running hybrid-computer programs.

The user types simple monitor commands, say "FORTRAN," and the monitor responds by loading the FORTRAN compiler and announces this fact with the CRT response FORTRAN. The user next types a command string comprising the name of the subject file to be compiled, plus options as to listings and commands to load and execute immediately. In the latter case, assuming successful compilation, the monitor will call a loader program which loads the binary object programs together with all required library routines. The monitor can also supply standard interrupt-servicing routines, keep track of the time expended on various computing jobs, control *chaining of core overlays from a disk for long programs* and, on command, *dump data from or onto tape files for later processing.*

Hybrid-computer library routines (Sec. 9-12) are ordinarily loaded from the disk as they are needed with compiled programs. A good hybrid monitor, though, must be able to access these routines in an interpreter mode as well, in order to permit "manual" program modifications (Sec. 9-12). The required interpreter program can be part of the resident monitor or, since core is usually at a premium, it may be quickly loaded on command. References 37 to 47 describe a number of such complete operating systems designed for large installations.

9-15. Use of Simulation Languages.[53-55] Most American simulation users are familiar with FORTRAN and demand its use for hybrid computation (with suitable real-time extensions, Sec. 9-13*e*). FORTRAN is well suited for controlling iterative-differential-analyzer programs, with each analog-computer run called by a FORTRAN subroutine (CALL RUN). But digital solution of differential equations

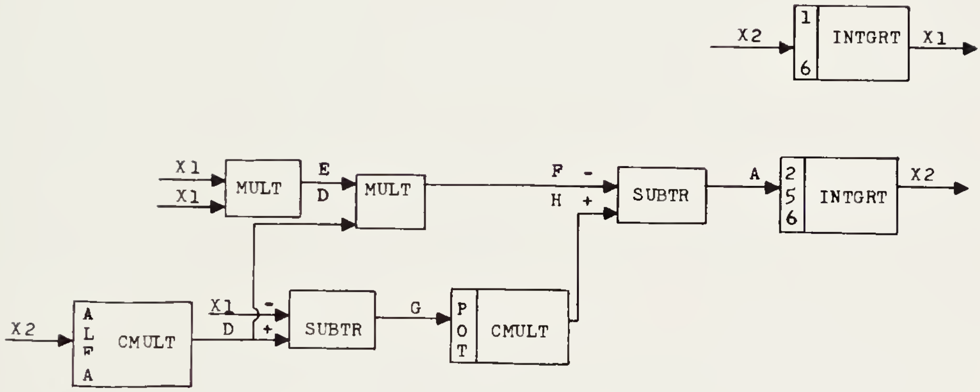
$$\dot{y} = f(t,y) \tag{9-12}$$

which is often needed for combined simulation, requires the user to write a routine which computes each derivative $f(t,y)$ (possibly several times, as for Runge-Kutta integration) and then combines increments $f(t,y)\,\Delta t$ according to some integration formula. In addition, the combined-simulation program requires data storage and output, plus an uncomfortable number of data type declarations, FORMAT, and DIMENSION statements.

Much programming labor not directly related to the simulation problem is eliminated by *continuous-system simulation languages*, which require the programmer merely to specify his differential-equation system either by a FORTRAN-like expression for each derivative or by statements describing interconnections of analog-computer-like blocks (Fig. 9-15). The user also enters the integration frame time Δt, the total computation time, and the integration method to be used. A simulation-language *translator* (*precompiler*) *program* will then automatically combine this derivative-computation routine with a suitable integration program. Simulation languages also perform multiple-run and iterative computations and can simplify

```
*DERIVATIVE BLOCK:

*VAN DER POL'S EQUATION

*NOTE TITLE AND COMMENTS FOR REPORT
*PREPARATION

X'=XDOT

XDOT'=ALFA*(1. - X*X)*XDOT - X

PHASE PLANE DISPLAY

DISPLAY XDOT, X


*DATA:

DT = 1.0E-01

TMAX = 2.0E+01

X = 1.

XDOT = 0.

ALFA = 1.
```

(a)



```
* DERIVATIVE BLOCK NO. 1:

/VAN DER POL'S EQUATION
       CMULT  D,X2,ALFA
       INTGRT X2,A,256
       CMULT  H,G,POT
       MULT   E,X1,X1
       INTGRT X1,X2,16
       MULT   F,E,D
       SUBTR  G,X1,D
       SUBTR  A,F,H
/RUN-TIME PHASE-PLANE DISPLAY
       DISPLAY X2,X1
```

```
* DATA:

DT = .0039
TMAX = .99
X2 = .5
X1 = -.5
ALFA = .9
POT = .0625
```

(b)

Fig. 9-15. Simulation-language programs for Van der Pol's differential equation (Sec. 2-11) in the form

$$\frac{dX}{dt} = \dot{X} \qquad \frac{d\dot{X}}{dt} = \alpha(1 - X^2)\dot{X} - X$$

Figure 9-15*a* uses a floating-point, equation-oriented language, which completely frees the user from scaling. Figure 9-15*b* uses a block-oriented language designed for very fast fixed-point computation on small digital computers and implements the scaled relations

$$\frac{dX_1}{dT} = 16X_2 \qquad \frac{dX_2}{dT} = 256[\alpha(1/16 - X_1^2)X_2 - 1/16 X_1]$$

where $$X_1 = \left(\frac{X}{4}\right) \qquad X_2 = \left(\frac{\dot{X}}{4}\right) \qquad T = \left(\frac{t}{16}\right)$$

range between 0 and 1 machine unit in absolute value (see also Ref. 55).

programming greatly by restricting data types to REAL variables and output to a few standard tabulation, display, and plotting options. Such languages are very widely used for all-digital simulation and are strongly recommended for combined analog-digital simulation as well.[55]

User preference for FORTRAN programming prevents application of small digital computers (without floating-point hardware) to combined simulation if more than a very small amount of digital computation is required during each frame time. This is a pity, for fixed-point *assembly-language* execution on very small machines (e.g., Digital Equipment Corporation PDP-15 or Data General SUPERNOVA) can match FORTRAN execution speeds of much larger digital computers in many simulation problems (small machines can easily perform six-degree-of-freedom flight simulation). For hybrid-computer simulation, fixed-point digital computation is not really very objectionable, since all analog variables must be scaled in any case. It is not practical to demand that users such as aircraft or chemical engineers learn assembly-language programming, but *it is possible to write convenient and powerful block-oriented simulation languages which generate efficient assembly-language programs.* The DARE II system illustrated by the example of Fig. 9-15*b*, for instance, has an optimizing precompiler which will not only order the user's block statements correctly but will also eliminate redundant instructions to produce code as efficient as that written by a good machine-language programmer.[55]

Macro-block simulation languages like that illustrated in Fig. 9-15*b* have also been successfully applied to the digital control of iterative and statistical computations with very fast analog computers. In such applications, the digital computer does not have to solve differential equations but performs parameter-optimization or statistics-accumulation routines between analog-computer runs. With "slow" analog computers requiring between 1 and 100 sec for each differential-equation-solving run, there is no objection to FORTRAN computation between runs even with small digital computers. But with very fast iterative differential analyzers such as LOCUST (Fig. 7-2), complete differential-equation-solving runs require only 0.25 to 1 msec, so that the digital processing time between analog-computer runs becomes critical.

9-16. Hybrid-computer Diagnostic Programs (see also Sec. 7-7). Special computer programs can automate *maintenance or diagnostic checks of analog computers and linkages.* With special problem boards in the analog and digital patchbays, such diagnostic programs exercise analog computer and linkage, measure suitably defined component errors, and print permanent maintenance records describing the condition of each computing element and pinpointing faulty elements. Such automated checks can be performed periodically and/or after computer malfunctions.

9-17. Time Sharing. Since, in most hybrid simulations, operators require time between simulation runs to look at solution records, enter new coefficients, etc., some sort of *time sharing* is mandatory for the efficient operation of all but the smallest digital computers used in hybrid computation. Even small computer installations can allow for a *background batch program* (or background listing, editing, etc.), which is interrupted by the foreground hybrid program at the latter's discretion. Larger computer laboratories may have exclusive use of very powerful digital computers (such as a CDC 6400 or 6600), which can be time-shared among four or more hybrid and other computation jobs. Because of the substantial programming and core overhead required to interrupt and swap time-shared programs in and out of core, such systems require careful economic justification. Time-shared hybrid-computer systems may either assign fixed periodic time slices to the individual simulation, or especially time-critical, high-priority programs may interrupt other programs.

The possibility of time sharing an *analog computer* through automatic patching and the software required for this purpose are discussed in Sec. 7-4*c*.

REFERENCES AND BIBLIOGRAPHY

Data Reconstruction

(See also Ref. 56)

1. Lofgren, L.: Predictors in Time-shared Analog Computer, *Proc. 1st AICA Conf.*, Brussels, 1955; Presses Académiques Européennes, Brussels, 1956.
2. Korn, G. A.: A Simple First-order-hold Circuit, *Simulation*, February 1964.
3. Wait, J. V.: Data Reconstruction: Fractional and Polygonal Holds, *Instr. Control Systems*, 1970.

Analysis and Correction of Errors due to Digital-computer Processing Delays

(See also Ref. 56)

4. Miura, T., and J. Iwata: Effects of Digital Execution Time in a Hybrid Computer, *Proc. FJCC*, 1963.
5. Gelman, R.: Corrected Inputs: A Method for Improved Hybrid Simulation, *Proc. FJCC*, 1963.
6. Gilbert, E. G.: Dynamic Error Analysis of Digital and Combined Analog-digital Computer Systems, *Simulation*, April 1966.
7. Bekey, G. A., and R. Tomovič: Adaptive Sampling Based on Amplitude Sensitivity, *IEEETAC*, April 1966.
8. Vichnevetsky, R.: Error Analysis in the Computer Simulation of Dynamic Systems: Variational Aspects of the Problem, *IEEETEC*, August 1967.

9. Mitchell, E. E.: The Effect of Digital Compensation for Computation Delay in a Hybrid Loop, *Proc. FJCC*, 1967.
10. Ingalls, V. W.: A Time-delay Compensation Technique for Hybrid Flight Simulators, *Boeing Memo M.S.* 23-68, Boeing Airplane Co., Seattle, Wash., 1968.
11. Giloi, W.: Error-corrected Operation of Hybrid Computer Systems, *Proc. 5th AICA Conf.*, Lausanne, 1967; Presses Académiques Européennes, Brussels, 1968.
12. Vichnevetsky, R.: Stability Contours for the Analysis of Analog/Digital Hybrid Simulation Loops, *Proc. SJCC*, 1969.
13. Kemp, N. H.: Extension and Analysis of Use of Derivatives for Compensation of Hybrid Solution of Linear Differential Equations, *Proc. FJCC*, 1969.
14. Dyntschaever, D.: Compensation of Time Delay and Smoothing Errors in Hybrid Computation Loops, *Electron. Letters*, May 1969.
15. VanSteenkiste, G. C.: Study of the Errors Introduced by Delays, *Ann. AICA*, October 1969.

Correction of Quantization Errors

(See also Ref. 56)

16. Vidal, J., and W. J. Karplus: Characterization and Compensation of Quantization Errors, *IEEE Natl. Convention Record*, 1965.
17. Miura, T., and I. Iwata: A Scheme for Reducing Quantization Errors in Hybrid-computer Systems, *Proc. 5th AICA Conf.*, Lausanne, 1967, Presses Académiques Européennes, Brussels, 1968.
18. Korn, G. A.: *Random-process Simulation and Measurements*, McGraw-Hill, New York, 1966.

Hybrid-computer Time-delay Simulation

19. Eide, M. A., and J. H. Forrester: A Hybrid Transport-delay Simulator, *Rept.* 2-5323-4/77, The Boeing Co., Seattle, Wash., 1963.
20. Nilsen, R. N.: Hybrid Simulation of a Variable Transport Lag, *Simulation*, February 1969.
21. Naka, M.: Data-channel Interface for Variable Time Delays, *CSRL Memo* 197, University of Arizona, 1970.

Hybrid Computers and Function Generators Employing Digital Computation with Analog Interpolation

(See also References to Sec. 6-14)

22. Jarrett, R. J.: An Analog *RC* Integrator with Digital Output, *Proc. NEC*, 1960.
23. Skramstad, H. K.: Combined Analog-Digital Differential Analyzer, *Proc. Eastern Joint Computer Conf.*, 1959.
24. Schmid, H.: Combined Analog-Digital Computing Elements, *Proc. Western Joint Computer Conf.*, 1961.
25. Korn, G. A.: The Impact of Hybrid Analog-Digital Techniques on the Analog-computer Art, *Proc. IRE*, May 1962.
26. Wait, J. V.: "A Hybrid Analog-Digital Differential-analyzer System," Ph.D. Thesis, University of Arizona, 1963; see also *Proc. Fall Joint Computer Conf.*, 1963.
27. O'Grady, P.: "A Simplified Hybrid Differential Analyzer," M.S. Thesis, University of Arizona, 1964; see also *Ann. AICA*, January 1966.
28. Tomovič, R., and W. J. Karplus: *High-speed Analog Computers*, Wiley, New York, 1962.
29. Schmid, H.: Linear-segment Function Generator, *IRETEC*, December 1962.
30. Chapelle, W. E.: Hybrid Techniques for Analog Function Generation, *Proc. Spring Joint Computer Conf.*, 1963.
31. Universal Nonlinear Element Model CI-230, *Instruction Manual*, Comcor, Inc., Denver, Colo., 1963.
32. Tutelman, D. M.: Computer Technique for Nonlinear-function Generation, *Simulation*, May 1966.
33. Ginzburg, S. A., and Y. Y. Lyubarskii: A Hybrid Function Generator, *Automatika i Telemekhanika*, August 1966.

34. Rubin, A. I.: Hybrid Techniques for Generation of Arbitrary Functions, *Simulation*, December 1966.
35. Aus, H. M., and G. A. Korn: Table-lookup/Interpolation Function Generation for Fixed-point Digital and Hybrid Computers, *IEETEC*, August 1969.
36. Giloi, W., and K. Waldschmidt: Neue Dimensionen für Genauigkeit und Geschwindigkeit analoger Rechenelemente, *Proc. 6th AICA Conf.*, Munich, 1970, Presses Académiques Européennes, Brussels, 1971.

Hybrid-computer Software

(See also Ref. 56)

37. Reed, E. F., et al.: An Integrated Software System for Hybrid Flight Simulation: The Boeing Hybrid Simulator, *Proc. IEEE*, December 1966.
38. Wilson, B. R.: The Boeing Integrated Hybrid Operating System, *Simulation*, November 1967.
39. Fineberg, M. S., and O. Serlin: Multiprogramming for Hybrid Computation, *Proc. FJCC*, 1967.
40. Jacoby, E. A., et al.: FAMILY I: Software for NASA/Ames Simulation Systems, *Proc. FJCC*, 1968.
41. Soma, G. N., et al.: A Priority-interrupt-oriented Hybrid Executive, *Proc. FJCC*, 1968.
42. Thompson, M. D.: Growing Pains in the Evolution of Hybrid Executives, *Proc. FJCC*, 1968.
43. Willard, D. A.: The Boeing/Vertoi Hybrid Executive System, *Proc. FJCC*, 1968.
44. Gutz, S.: An Extension of a FORTRAN Compiler for Hybrid Computation, *Proc. SJCC*, 1969.
45. Graves, W. L., and R. A. MacDonald: Hybrid Executive: A User's Approach, *Proc. FJCC*, 1969.
46. Bedient, C. K., and L. L. Dike: The Lockheed Hybrid System, *Proc. FJCC*, 1969.
47. Franklin, M. A., and J. C. Strauss: A Hybrid-computer Programming System, *Proc. FJCC*, 1969.
48. Benham, R. D., et al.: SIMPL-1, A Simple Approach to Simulation, *Simulation*, September 1969.
49. Serlin, O., and R. C. Gerard: Remote Real-time Simulation, *Proc. SJCC*, 1970.
50. Dodds, W. R.: Multiprogramming in a Medium-sized Hybrid Environment, *Proc. FJCC*, 1970.
51. Pelle, J.: Hybrid-computer Diagnostics, *Proc. 6th AICA Conf.*, Munich, 1970, Presses Académiques Européennes, Brussels, 1971.
52. Giloi, W. K.: A Standard Programming System for Hybrid Computation and Its Implementation, *MIT Electron. Systems Lab. Rept.* ESL-R-427, Cambridge, Mass., 1970; see also *Proc. SJCC*, 1969.

Use of Simulation Languages

53. Brennan, R. D., and R. N. Linebarger: A Survey of Digital Simulation, *Simulation*, December 1964.
54. Tossman, B. E., et al.: SIMCON: An Advancement in Simulation of Physical Systems, *Proc. FJCC*, 1970.
55. Korn, G. A.: New Techniques for All-digital and Hybrid Simulation, *Proc. 6th AICA Conf.*, Munich, 1970, Presses Académiques Européennes, Brussels, 1971.

Hybrid-computer Diagnostic Programs

(See also references to Sec. 7-7)

56. Richards, J. C.: Automatic Checkout of a Large Hybrid System, *Proc. FJCC*, 1968.
57. Seehuns, T. K., et al.: Hybrid Diagnostic Techniques, *Proc. FJCC*, 1968.

Miscellaneous

58. Bekey, G. A., and W. J. Karplus: *Hybrid Computation*, Wiley, New York, 1968.
59. Connelly, M. E.: A Demonstration Hybrid Computer for Real-time Flight Simulation, *Simulation*, September 1965.

Function-fitting Methods

60. Goldhirsh, I. L.: Computation of Continuous Functions, *Simulation*, July 1968.
61. Van Steenkiste, G. J. V.: Optimal Generation of Arbitrary Functions, *Ann. AICA*, July 1967 (see also review by E. J. Copesin, *IEEETEC*, July 1968).
62. Gracon, T. J., and J. C. Strauss: Application of Dynamic Programming to Function Generation, *Simulation*, November 1968.

APPENDIX A

OPERATIONAL-AMPLIFIER DESIGN DATA

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers*

(Sec. 1-10; see Secs. 3-10 to 3-17 for error analysis)

This table is used to find networks whose transfer impedances $Z_0(P), Z_1(P), Z_2(P), \ldots,$ match the desired performance equation

$$X_0 = -\frac{Z_0(P)}{Z_1(P)}X_1 \qquad \text{or} \qquad X_0 = -Z_0(P)\left[\frac{X_1}{Z_1(P)} + \frac{X_2}{Z_2(P)} + \cdots\right]$$

for an operational amplifier of one of the types shown in Fig. 1-7*a*, *b*, *c*, *d*. See Fig. 1-9 for an example illustrating the use of the table.

Short-circuit transfer impedances for many additional networks can be derived with the aid of the following theorems:

1. *Two or more networks may be paralleled to yield a new transfer impedance*

$$Z(P) = \left[\frac{1}{Z_1(P)} + \frac{1}{Z_2(P)} + \cdots\right]^{-1}$$

2. *For every (asymmetrical as well as symmetrical) T network, the output-input transfer impedance is identical with the input-output transfer impedance given in the table.*

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers* (*Continued*)

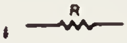
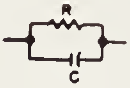
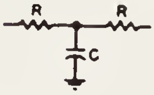
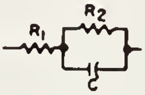
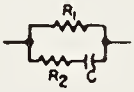
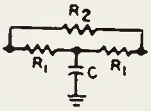
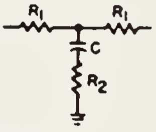
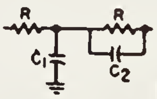
Transfer-impedance function	Network	Relations	Inverse relations
A		$A = R$	$R = A$
$\frac{A}{1 + PT}$		$A = R$ $T = RC$	$R = A$ $C = \frac{T}{A}$
$A(1 + PT)$		$A = 2R$ $T = \frac{RC}{2}$	$R = \frac{A}{2}$ $C = \frac{4T}{A}$
$A\left(\frac{1 + P\theta T}{1 + PT}\right)$ $\theta < 1$		$A = R_1 + R_2$ $T = R_2C$ $\theta = \frac{R_1}{R_1 + R_2}$	$R_1 = A\theta$ $R_2 = A(1 - \theta)$ $C = \frac{T}{A(1 - \theta)}$
		$A = R_1$ $T = (R_1 + R_2)C$ $\theta = \frac{R_2}{R_1 + R_2}$	$R_1 = A$ $R_2 = \frac{A\theta}{1 - \theta}$ $C = \frac{T(1 - \theta)}{A}$
$A\left(\frac{1 + PT}{1 + P\theta T}\right)$ $\theta < 1$		$A = \frac{2R_1R_2}{2R_1 + R_2}$ $T = \frac{R_1C}{2}$ $\theta = \frac{2R_1}{2R_1 + R_2}$	$R_1 = \frac{A}{2(1 - \theta)}$ $R_2 = \frac{A}{\theta}$ $C = \frac{4T(1 - \theta)}{A}$
		$A = 2R_1$ $T = \left(R_2 + \frac{R_1}{2}\right) C$ $\theta = \frac{2R_2}{2R_2 + R_1}$	$R_1 = \frac{A}{2}$ $R_2 = \frac{A\theta}{4(1 - \theta)}$ $C = \frac{4T(1 - \theta)}{A}$
		$A = 2R$ $T = \frac{R}{2}(C_1 + C_2)$ $\theta = \frac{2C_2}{C_1 + C_2}$	$R = \frac{A}{2}$ $C_1 = \frac{2T(2 - \theta)}{A}$ $C_2 = \frac{2T\theta}{A}$

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers* (*Continued*)

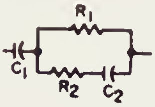
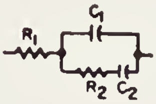
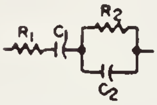
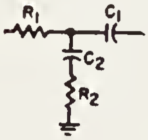
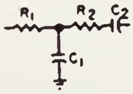
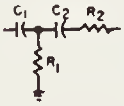
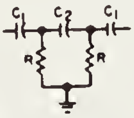
Transfer-impedance function	Network	Relations	Inverse relations
$\frac{1}{PB}\left[\frac{(1+PT_1)(1+PT_3)}{1+PT_2}\right]$ $T_1 < T_2 < T_3$		$B = C_1$ $T_2 = (R_1 + R_2)C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \frac{T_1 + T_3 - T_2}{B}$ $R_2 = \frac{T_1T_3(T_1 + T_3 - T_2)}{B(T_3 - T_2)(T_2 - T_1)}$ $C_1 = B$ $C_2 = \frac{B(T_3 - T_2)(T_2 - T_1)}{(T_1 + T_3 - T_2)^2}$
		$B = C_1 + C_2$ $T_2 = R_2\left(\frac{C_1C_2}{C_1 + C_2}\right)$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \frac{T_1T_3}{BT_2}$ $R_2 = \frac{(T_1T_2 + T_2T_3 - T_1T_3)^2}{BT_2(T_3 - T_2)(T_2 - T_1)}$ $C_1 = \frac{BT_2^2}{T_1T_2 + T_2T_3 - T_1T_3}$ $C_2 = \frac{B(T_3 - T_2)(T_2 - T_1)}{T_1T_2 + T_2T_3 - T_1T_3}$
		$B = C_1$ $T_2 = R_2C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = \frac{T_1T_3}{BT_2}$ $R_2 = \frac{(T_3 - T_2)(T_2 - T_1)}{BT_2}$ $C_1 = B$ $C_2 = \frac{BT_2^2}{(T_3 - T_2)(T_2 - T_1)}$
		$B = C_1$ $T_2 = R_2C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \frac{T_1T_3}{BT_2}$ $R_2 = \frac{T_1T_2T_3}{B(T_3 - T_2)(T_2 - T_1)}$ $C_1 = B$ $C_2 = \frac{B(T_3 - T_2)(T_2 - T_1)}{T_1T_3}$
$\frac{1}{PB}(1 + PT_1)(1 + PT_2)$ $T_1 \neq T_2$		$B = C_2$ $T_1T_2 = R_1R_2C_1C_2$ $T_1 + T_2 = R_1C_1 + R_2C_2 + R_1C_2$	$R_1 = \frac{(\sqrt{T_1} - \sqrt{T_2})^2}{B}$ $R_2 = \frac{\sqrt{T_1T_2}}{B}$ $C_1 = \frac{B\sqrt{T_1T_2}}{(\sqrt{T_1} - \sqrt{T_2})^2}$ $C_2 = B$
$\frac{1}{PB}\left[\frac{(1+PT_1)(1+PT_2)}{P\sqrt{T_1T_2}}\right]$ $T_1 \neq T_2$		$B = C_2$ $T_1T_2 = R_1R_2C_1C_2$ $T_1 + T_2 = R_1C_1 + R_2C_2 + R_1C_2$	$R = \frac{(\sqrt{T_1} - \sqrt{T_2})^2}{B}$ $R_2 = \frac{\sqrt{T_1T_2}}{B}$ $C_1 = \frac{B\sqrt{T_1T_2}}{(\sqrt{T_1} - \sqrt{T_2})^2}$ $C_2 = B$
$\frac{1}{PB}\left[\frac{(1+PT_1)(1+PT_2)}{P^2T_1T_2}\right]$ $T_1 < T_2$		$B = \frac{C_1C_2}{C_1 + 2C_2}$ $T_1 = RC_1$ $T_2 = R(C_1 + 2C_2)$	$R = \frac{T_1(T_2 - T_1)}{2BT_2}$ $C_1 = \frac{2BT_2}{T_2 - T_1}$ $C_2 = \frac{BT_2}{T_1}$

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers* (*Continued*)

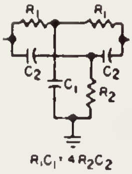
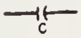
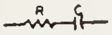
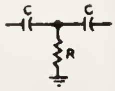
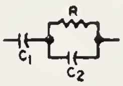
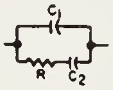
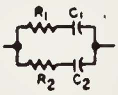
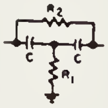
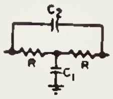
Transfer-impedance function	Network	Relations	Inverse relations
$A\left(\frac{1+PT_1}{1+P^2T_1T_2}\right)$		$A = 2R_1$ $T_1 = \frac{R_1C_1}{2} = 2R_2C_2$ $T_2 = R_1C_2$	$R_1 = \frac{A}{2}$ $R_2 = \frac{AT_1}{4T_2}$ $C_1 = \frac{4T_1}{A}$. $C_2 = \frac{2T_2}{A}$
$\frac{1}{PB}$		$B = C$	$C = B$
$\frac{1}{PB}(1+PT)$		$B = C$ $T = RC$	$R = \frac{T}{B}$ $C = B$
$\frac{1}{PB}\left(\frac{1+PT}{PT}\right)$		$B = \frac{C}{2}$ $T = 2RC$	$R = \frac{T}{4B}$ $C = 2B$
$\frac{1}{PB}\left(\frac{1+PT}{1+P\theta T}\right)$ $\theta < 1$		$B = C_1$ $T = R(C_1 + C_2)$ $\theta = \frac{C_2}{C_1 + C_2}$	$R = \frac{T(1-\theta)}{B}$ $C_1 = B$ $C_2 = \frac{B\theta}{1-\theta}$
		$B = C_1 + C_2$ $T = RC_2$ $\theta = \frac{C_1}{C_1 + C_2}$	$R = \frac{T}{B(1-\theta)}$ $C_1 = B\theta$ $C_2 = B(1-\theta)$
$\frac{1}{PB}\left[\frac{(1+PT_1)(1+PT_3)}{1+PT_2}\right]$ $T_1 < T_2 < T_3$		$B = C_1 + C_2$ $T_1 = R_1C_1$ $T_2 = (R_1 + R_2)\left(\frac{C_1C_2}{C_1 + C_2}\right)$ $T_3 = R_2C_2$	$R_1 = \frac{T_1(T_3 - T_1)}{B(T_2 - T_1)}$ $R_2 = \frac{T_3(T_3 - T_1)}{B(T_3 - T_2)}$ $C_1 = \frac{B(T_2 - T_1)}{T_3 - T_1}$ $C_2 = \frac{B(T_3 - T_2)}{T_3 - T_1}$
$A\left(\frac{1+PT_1}{1+PT_1+P^2T_1T_2}\right)$		$A = R_2$ $T_1 = 2R_1C$ $T_2 = \frac{R_2C}{2}$	$R_1 = \frac{AT_1}{4T_2}$ $R_2 = A$ $C = \frac{2T_2}{A}$
$A\left(\frac{1+PT_2}{1+PT_1+P^2T_1T_2}\right)$		$A = 2R$ $T_1 = 2RC_2$ $T_2 = \frac{RC_1}{2}$	$R = \frac{A}{2}$ $C_1 = \frac{4T_2}{A}$ $C_2 = \frac{T_1}{A}$

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers* (*Continued*)

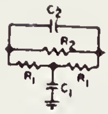
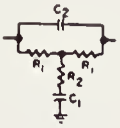
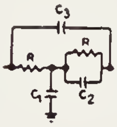
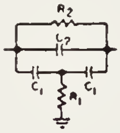
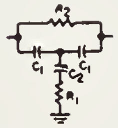
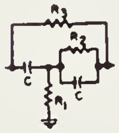
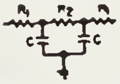
Transfer-impedance function	Network	Relations	Inverse relations
$A\left[\frac{1+PT_3}{1+PT_1+P^2T_1T_2}\right]$ $T_2 > \frac{T_1}{4}$ (complex roots) $T_3 > T_2$		$A = \frac{2R_1R_2}{(2R_1+R_2)}$ $T_1 = \frac{R_1(R_1C_1+2R_2C_2)}{2R_1+R_2}$ $T_2 = \frac{R_1R_2C_1C_2}{R_1C_1+2R_2C_2}$ $T_3 = \frac{R_1C_1}{2}$	$R_1 = \frac{AT_3^2}{2[T_3^2-T_1(T_3-T_2)]}$ $R_2 = \frac{AT_3^2}{T_1(T_3-T_2)}$ $C_1 = \frac{4[T_3^2-T_1(T_3-T_2)}{AT_3}$ $C_2 = \frac{T_1T_2}{AT_3}$
		$A = 2R_1$ $T_1 = R_2C_1 + 2R_1C_2$ $T_2 = \frac{R_1(R_1+2R_2)C_1C_2}{R_2C_1+2R_1C_2}$ $T_3 = \left(R_2+\frac{R_1}{2}\right)C_1$	$R_1 = \frac{A}{2}$ $R_2 = \frac{AT_1(T_3-T_2)}{4[T_3^2-T_1(T_3-T_2)]}$ $C_1 = \frac{4[T_3^2-T_1(T_3-T_2)]}{AT_3}$ $C_2 = \frac{T_1T_2}{AT_3}$
		$A = 2R$ $T_1 = R(C_2 + 2C_3)$ $T_2 = \frac{RC_3(C_1+C_2)}{C_2+2C_3}$ $T_3 = \frac{R}{2}(C_1+C_2)$	$R = \frac{A}{2}$ $C_1 = \frac{2[2T_3^2-T_1(T_3-T_2)]}{AT_3}$ $C_2 = \frac{2T_1(T_3-T_2)}{AT_3}$ $C_3 = \frac{T_1T_2}{AT_3}$
$A'\left[\frac{1+PT_3}{1+PT_1+P^2T_1T_2}\right]$ $T_2 > \frac{T_1}{4}$ (complex roots) $T_3 < T_1$		$A = R_2$ $T_1 = 2R_1C_1 + R_2C_2$ $T_2 = \frac{R_1R_2C_1(C_1+2C_2)}{2R_1C_1+R_2C_2}$ $T_3 = 2R_1C_1$	$R_1 = \frac{AT_3^2}{4[T_1T_2-T_3(T_1-T_3)]}$ $R_2 = A$ $C_1 = \frac{2[T_1T_2-T_3(T_1-T_3)]}{AT_3}$ $C_2 = \frac{(T_1-T_3)}{A}$
		$A = R_2$ $T_1 = \frac{C_1(2R_1C_2+R_2C_1)}{2C_1+C_2}$ $T_2 = \frac{R_1R_2C_1C_2}{2R_1C_2+R_2C_1}$ $T_3 = \frac{2R_1C_1C_2}{(2C_1+C_2)}$	$R_1 = \frac{AT_3^2}{4[T_1T_2-T_3(T_1-T_3)]}$ $R_2 = A$ $C_1 = \frac{2T_1T_2}{AT_3}$ $C_2 = \frac{4T_1T_2[T_1T_2-T_3(T_1-T_3)]}{AT_3^2(T_1-T_3)}$
		$A = R_3$ $T_1 = \frac{R_1(2R_2+R_3)C}{R_1+R_2}$ $T_2 = \frac{R_2R_3C}{(2R_2+R_3)}$ $T_3 = \frac{2R_1R_2C}{(R_1+R_2)}$	$R_1 = \frac{AT_3^2}{2[2T_1T_2-T_3(T_1-T_3)]}$ $R_2 = \frac{AT_3}{2(T_1-T_3)}$ $R_3 = A$ $C = \frac{2T_1T_2}{AT_3}$
$A(1+PT_1)(1+PT_2)$ $T_1 < T_2$		$A = 2R_1 + R_2$ $T_1 = \left(\frac{R_1R_2}{2R_1+R_2}\right)C$ $T_2 = R_1C$	$R_1 = A\left(\frac{T_2-T_1}{2T_2}\right)$ $R_2 = A\frac{T_1}{T_2}$ $C = \frac{2T_2^2}{A(T_2-T_1)}$

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers* (*Continued*)

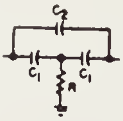
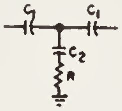
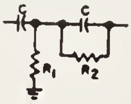
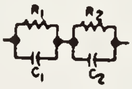
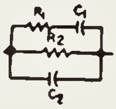
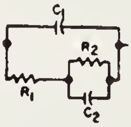
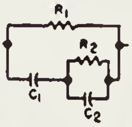
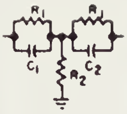
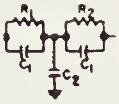
Transfer-impedance function	Network	Relations	Inverse relations
$\frac{1}{PB}\left(\frac{1+P\theta T}{1+PT}\right)$ $\theta < 1$		$B = C_2$ $T = RC_1\left(\frac{2C_2 + C_1}{C_2}\right)$ $\theta = \frac{2C_2}{2C_2 + C_1}$	$R = \frac{T\theta^2}{4B(1-\theta)}$ $C_1 = \frac{2B(1-\theta)}{\theta}$ $C_2 = B$
		$B = \frac{C_1^2}{2C_1 + C_2}$ $T = RC_2$ $\theta = \frac{2C_1}{2C_1 + C_2}$	$R = \frac{T\theta^2}{4B(1-\theta)}$ $C_1 = \frac{2B}{\theta}$ $C_2 = \frac{4B(1-\theta)}{\theta^2}$
		$B = \left(\frac{R_1}{R_1 + R_2}\right) C$ $T = R_2C$ $\theta = \frac{2R_1}{R_1 + R_2}$	$R_1 = \frac{T\theta^2}{2B(2-\theta)}$ $R_2 = \frac{T\theta}{2B}$ $C = \frac{2B}{\theta}$
$A\left[\frac{1 + PT_2}{(1 + PT_1)(1 + PT_3)}\right]$ $T_1 < T_2 < T_3$		$A = R_1 + R_2$ $T_1 = R_1C_1$ $T_2 = \left(\frac{R_1R_2}{R_1+R_2}\right)(C_1+C_2)$ $T_3 = R_2C_2$	$R_1 = \frac{A(T_2 - T_1)}{T_3 - T_1}$ $R_2 = \frac{A(T_3 - T_2)}{T_3 - T_1}$ $C_1 = \frac{T_1(T_3 - T_1)}{A(T_2 - T_1)}$ $C_2 = \frac{T_3(T_3 - T_1)}{A(T_3 - T_2)}$
		$A = R_2$ $T_2 = R_1C_1$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = \frac{AT_2^2}{(T_3 - T_2)(T_2 - T_1)}$ $R_2 = A$ $C_1 = \frac{(T_3 - T_2)(T_2 - T_1)}{AT_2}$ $C_2 = \frac{T_1T_3}{AT_2}$
		$A = R_1 + R_2$ $T_2 = \left(\frac{R_1R_2}{R_1 + R_2}\right) C_2$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = \frac{AT_2^2}{T_1T_2 + T_2T_3 - T_1T_3}$ $R_2 = \frac{A(T_3 - T_2)(T_2 - T_1)}{T_1T_2 + T_2T_3 - T_1T_3}$ $C_1 = \frac{T_1T_3}{AT_2}$ $C_2 = \frac{(T_1T_2 + T_2T_3 - T_1T_3)^2}{AT_2(T_3 - T_2)(T_2 - T_1)}$
		$A = R_1$ $T_2 = R_2(C_1 + C_2)$ $T_1T_3 = R_1R_2C_1C_2$ $T_1 + T_3 = R_1C_1 + R_2C_2 + R_2C_1$	$R_1 = A$ $R_2 = \frac{A(T_3 - T_2)(T_2 - T_1)}{(T_1 + T_3 - T_2)^2}$ $C_1 = \frac{T_1 + T_3 - T_2}{A}$ $C_2 = \frac{T_1T_3(T_1 + T_3 - T_2)}{A(T_3 - T_2)(T_2 - T_1)}$

Table A-1. Short-circuit Transfer Impedances for Operational Amplifiers* (*Continued*)

Transfer-impedance function	Network	Relations	Inverse relations
$A\left[\frac{1+PT_2}{(1+PT_1)(1+PT_3)}\right]$ $T_2 \leq T_1 \leq T_3$		$A = 2R_1 + \frac{R_1^2}{R_2}$ $T_1 = R_1C_1$ $T_2 = \left(\frac{R_1R_2}{R_1+2R_2}\right)(C_1+C_2)$ $T_3 = R_1C_2$	$R_1 = \frac{AT_2}{(T_1+T_3)}$ $R_2 = \frac{AT_2^2}{(T_1+T_3)(T_1+T_3-2T_2)}$ $C_1 = \frac{T_1(T_1+T_3)}{AT_2}$ $C_2 = \frac{T_3(T_1+T_3)}{AT_2}$
$A\left[\frac{1+PT_2}{(1+PT_1)(1+PT_3)}\right]$ $T_1 \leq T_3 \leq T_2$		$A = R_1 + R_2$ $T_1 = R_1C_1$ $T_2 = \frac{R_1R_2}{R_1+R_2}(2C_1+C_2)$ $T_3 = R_2C_1$	$R_1 = \frac{AT_1}{(T_1+T_3)}$ $R_2 = \frac{AT_3}{(T_1+T_3)}$ $C_1 = \frac{(T_1+T_3)}{A}$ $C_2 = \frac{(T_1+T_3)}{A}\left(\frac{T_2}{T_3}+\frac{T_2}{T_1}-2\right)$

---

* Reproduced by permission of the McGraw-Hill Publishing Company from F. R. Bradley and R. McCoy, Driftless D-c Amplifier, *Electronics*, April, 1952. This table was developed by S. Godet of the Reeves Instrument Corporation, New York City.

$$\frac{Y}{X} = \frac{-1}{\frac{1}{\omega_n^2}P^2 + \frac{2\xi}{\omega_n}P + 1}$$

$$\omega_n = \sqrt{\frac{1}{R_1 R_2 C_1 C_2}}$$

$$\xi = \left(\frac{R_1}{2} + R_2\right)\sqrt{\frac{C_2}{C_1}\frac{1}{R_1 R_2}} \geq \sqrt{\frac{2C_2}{C_1}}$$

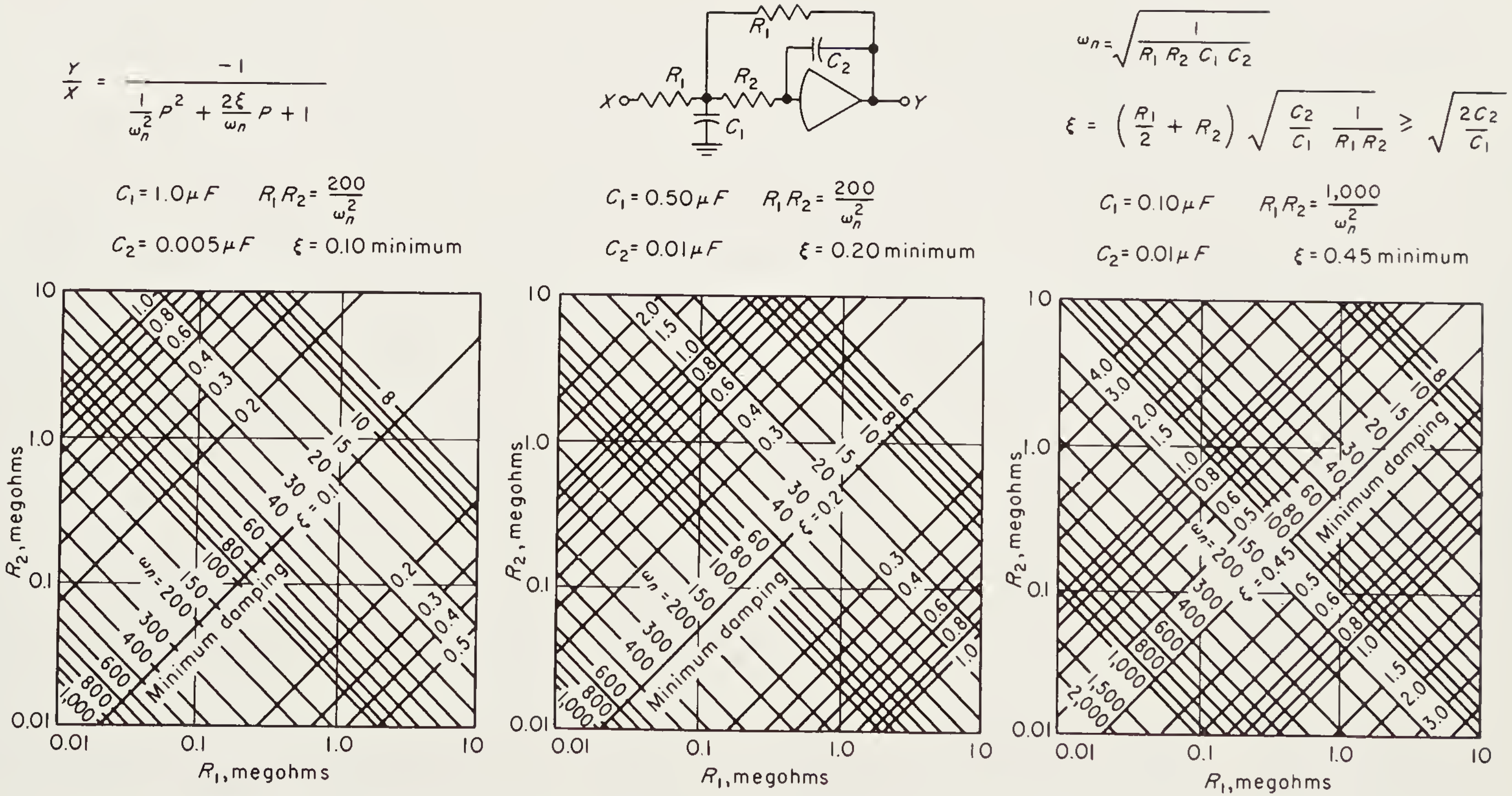


Fig. A-1. Design nomographs for a simple circuit permitting economical computer representation of many mass-spring-dashpot or equivalent systems (values, pressure pickups, rate gyros, accelerometers, etc.). (*Nomographs prepared by H. J. Shear, Ryan Aeronautical Co., San Diego, Calif., for G. W. Smith and R. C. Wood, Principles of Analog Computation, McGraw-Hill Book Company, New York,* 1959; *reproduced by permission of the publishers.*)

APPENDIX **B**

OPERATIONAL-AMPLIFIER CIRCUITS FOR TIME-DELAY SIMULATION

B-1. Introduction. For *constant* delay τ_D and suitably differentiable input $X(\tau)$, Taylor expansion yields the operator relation

$$Y(\tau) = X(\tau - \tau_D) = \left[1 - \tau_D P + \frac{1}{2!}(\tau_D P)^2 \mp \cdots\right] X(\tau)$$

$$= e^{-\tau_D P} X(\tau) \qquad \left(P \equiv \frac{d}{d\tau}\right) \qquad \text{(B-1)}$$

For steady-state sinusoidal input and output, the *constant-time-delay operator* $e^{-\tau_D P}$ corresponds to constant unity gain and negative phase shift (lag) proportional to the frequency f:

$$|e^{-j\omega\tau_D}| = 1 \qquad \arg e^{-j\omega\tau_D} = -\omega\tau_D = -2\pi f\tau_D \qquad \text{(B-2)}$$

At reasonably low frequencies, we can approximate $e^{-\tau_D s}$ by a rational transfer function $H(s)$ implemented with operational amplifiers and/or passive linear networks. In view of Eq. (2), correct phase shift at the frequency f will require at least $2f\tau_D$ energy-storing network elements (capacitors or inductors), since the phase shift contributed by each can increase (from zero at d-c) to at most 180 deg even in active networks.[9]

Table B-1 shows differential-analyzer setups for constant-delay approximations of the form

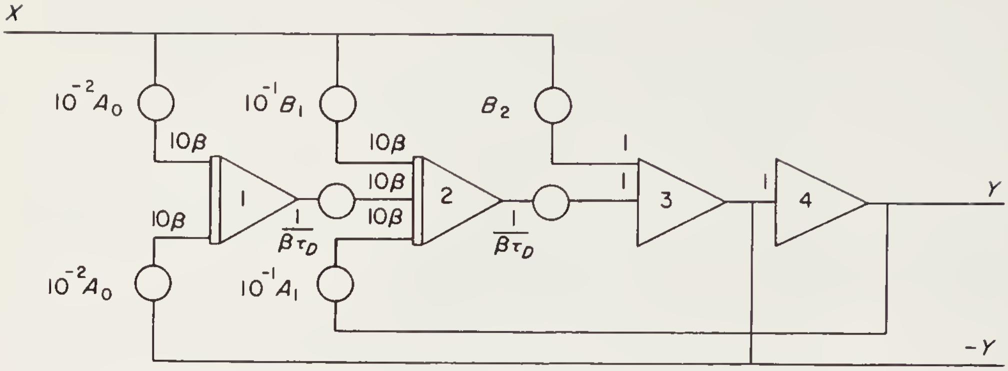
$$H(P) = \frac{A_0 - B_1(\tau_D P) + B_2(\tau_D P)^2 \mp \cdots + (-1)^n B_n(\tau_D P)^n}{A_0 + A_1(\tau_D P) + A_2(\tau_D P)^2 + \cdots + A_n(\tau_D P)^n}$$

$$\left(P \equiv \frac{d}{d\tau}\right) \qquad \text{(B-3)}$$

with positive coefficients (see also Sec. 1-8). Several different synthesis techniques[6] yield suitable coefficients A_i, B_k; the choice will depend on the nature of our signals and on the objective of the simulation.

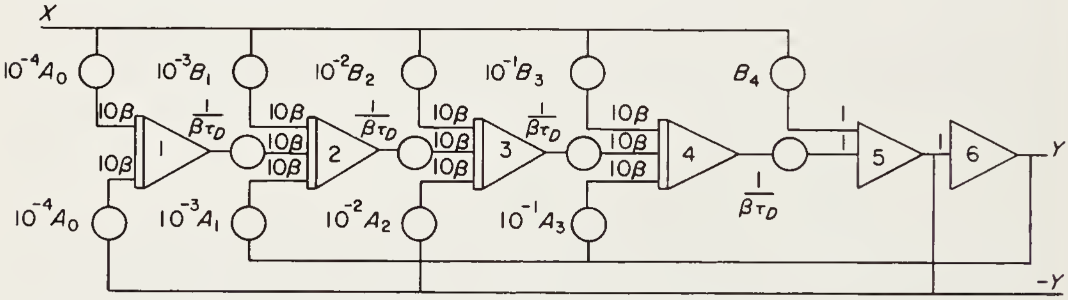
B-2. All-pass Approximations. With input signals $X(\tau)$ limited in frequency below about $1/\tau_D$ Hz (e.g., filtered signals and noise in control-system simulation), we can use *all-pass (zero-attenuation) approximations.* For $B_k = A_k$ ($k = 1, 2, \ldots, n$), Eq. (3) yields the desired unity gain $|H(j\omega)| = 1$. The A_k are then chosen to produce low phase error $\delta(\omega) = \arg H(j\omega) + \omega\tau_D$. Referring to Table B-1, *all-pass Padé approximations*[6] $P_{n,n}$

Table B-1. Time-delay Approximations



1. Second-order and Cascaded Second-order All-pass Delay Approximations (Primed quantities refer to the second network of a cascade circuit)

	A_0	$A_1 = B_1$	A'_0	$A'_1 = B'_1$	$\omega\tau_D$ limit for $\delta(\omega) < 0.1$ radian
All-pass Padé,[6] $P_{2,2}$	12	6	.....	....	2.9
Smith and Wood,[1] $W_{2,2}$	10	4.7	.....	....	3.4
Cunningham,[6] $C_{4,4}$	30.88	10.56	45.64	7.48	6.9
Stubbs and Single,[5] $S_{4,4}$	19.4	7.63	54.9	5.94	7.6



2. Fourth-order Delay Approximations

	A_0	A_1	A_2	A_3	B_1	B_2	B_3	B_4
All-pass Padé,[6] $P_{4,4}$	1,680	840	180	20	840	180	20	1
All-pass Single-type,[2] $S'_{4,4}$	1,074	537	120	13.59	536	120	13.55	1
Low-pass Padé,[6] $P_{3,4}$	840	480	120	16	360	60	4	0
King and Rideout,[6] $K_{3,4}$	537	361.2	101.9	12.98	175.8	22.45	1.51	0
King and Rideout,[6] $R_{3,4}$	476	309	91.8	12.1	160.5	22.8	1.48	0

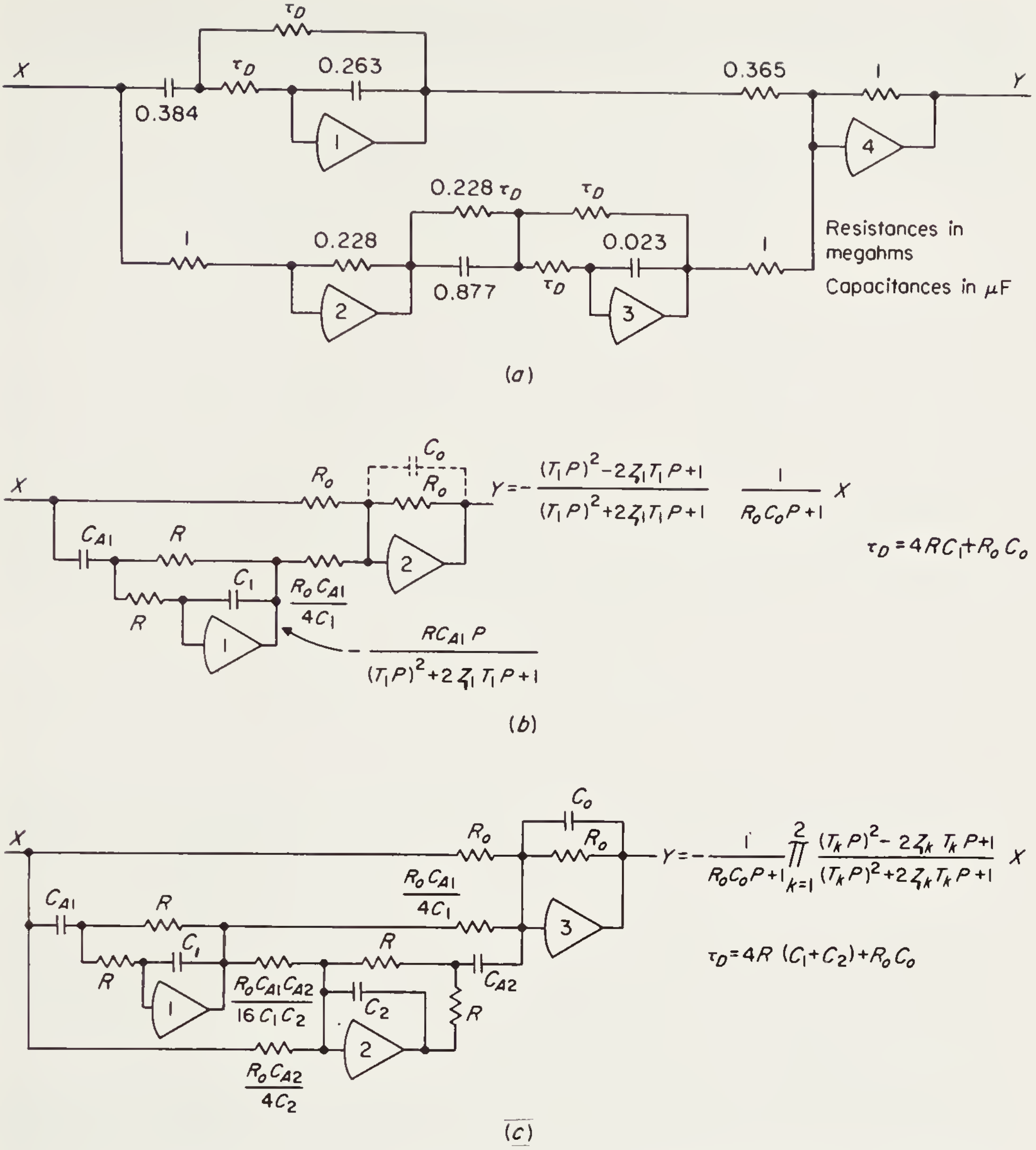


Fig. B-1. A four-amplifier circuit for $K_{3,4}$ (Table B-1) (*a*), and single-type operational-amplifier circuits (*b*), (*c*). A-c coupling can save chopper stabilization in some amplifiers. With chopper stabilization, circuit (*c*) tends to be noisy for low delays τ_D, so that two cascaded circuits (*b*) are preferable. We have

$$T_1 = R\sqrt{C_1C_{A1}} \qquad T_2 = R\sqrt{C_2C_{A2}} \qquad \zeta_1 = \sqrt{\frac{C_1}{C_{A1}}} \qquad \zeta_2 = \sqrt{\frac{C_2}{C_{A2}}}$$

and preferred values are given by

$$\zeta_1 = \sqrt{3}/2 \qquad \zeta_2 = 0.4 \qquad T_1/T_2 = 1.68 \qquad R_0C_0 \leq \tau_D/100$$

(*Based on Refs.* 5 *and* 6.)

yield a monotonically increasing phase error which is very small at low frequencies. A number of other all-pass approximations[5–14] alternate positive and negative phase errors to reduce the error near $f = 1/\tau_D$ at the expense of low-frequency phase errors; this is usually preferable.

In practice, advanced synthesis is often replaced by simple curve fitting.[5] All-pass approximants of even order $n = 2m$ are written in the form

$$\left.\begin{aligned} (HP) &= \prod_{k=1}^{m} \frac{(T_kP)^2 - 2\zeta_k T_k P + 1}{(T_kP)^2 + 2\zeta_k T_k P + 1} \qquad \left(P \equiv \frac{d}{d\tau}\right) \\ \text{with} \qquad \tau_D &= 4 \sum_{k=1}^{m} \zeta_k T_k \end{aligned}\right\} \tag{B-4}$$

The phase error is given by

$$\delta(\omega) = \sum_{k=1}^{m} \delta_k(\omega) \qquad \delta_k(\omega) = 4\zeta_k T_k \omega - 2 \arctan \frac{2\zeta_k T_k \omega}{1 - \omega^2 T_k^2} \tag{B-5}$$

Carefully plotted graphs[5,6] of 2 arctan $[2\zeta T\omega/(1 - \omega^2 T^2)]$ vs. $4\zeta T\omega$ are used to match the individual terms $\delta_k(\omega)$ for a small overall error $\delta(\omega)$. $\delta_k(\omega)$ will be positive for low ω if and only if $\zeta_k < \sqrt{3}/2$. For $m = 1$ $(n = 2)$, $\zeta_1 = \sqrt{3}/2$ yields the all-pass Padé approximant $P_{2,2}$, and $\zeta = 0.8165$ results in a useful approximation with initially negative error. A good fourth-order approximant ($m = 2$, $n = 4$) is obtained with

$$\zeta_1 = \frac{\sqrt{3}}{2} \qquad \zeta_2 = 0.4 \qquad \frac{T_1}{T_2} = 1.68$$

(Table B-1 and Fig. B-1).[5] Reference 1 also gives sixth-order and eighth-order approximations.

B-3. Low-pass Approximations. If our signals are not band-limited (e.g., step inputs) then we require *low-pass delay approximations*, since we must attenuate high-frequency components which cannot be correctly delayed. Such high-frequency components will, in particular, cause unpleasant transient oscillations in the step response (Fig. B-2*b*, *c*). Delay-approximation synthesis methods are based on different compromises between good approximations to the ideal frequency response (2) and clean step response (e.g., minimum mean-square step-response error).[5,6] Several synthesis techniques simply combine an all-pass delay network with a low-pass filter (Fig. B-1*c*).

Table B-1 gives several low-pass approximations of the form (3) with $B_n = 0$, and Fig. B-2 illustrates typical frequency and step responses.

B-4. Practical Considerations. Treatment of Variable Time Delays. Several similar delay networks can be cascaded for higher-order approximations to longer delays, although this will not yield the best high-order approximants. Approximations of order $n > 4$ are, in any case, best implemented as combinations of (in general, different) lower-order networks to avoid computer-setup instability. The general differential-analyzer setups of Table B-1 can be simplified if we do not need to vary the delay τ_D frequently (Fig. B-1*a*).

For permanently assembled time-delay elements, we can save amplifiers as well as potentiometers with special operational-amplifier setups.[5,6] Commercially available units of this type employ low-pass circuits similar to those of Fig. B-1 to obtain adjustable constant delays between 0.005 and 10 sec with switched 20:1 ranges. Two or three channels can be cascaded for longer delays.

If the delay time τ_D is variable, representation of $Y(\tau) = X[\tau - \tau_D(\tau)]$ in the form (1) requires caution. Since now $\tau_D P \neq P\tau_D$, the $1/\beta\tau_D$ potentiometers in Table B-1 must be replaced by summers and multipliers (or dividers) *preceding* each integrator.[9]

In many variable-flow transport-delay situations, the delay $\tau_D(\tau)$ varies as a nondecreasing or nonincreasing differentiable function of the time τ. In such cases, it may be practical to introduce a new independent variable $q(\tau)$ such that

$$X(\tau) = \xi(q) \qquad Y(\tau) = X[\tau - \tau(\tau_D)] = \xi(q - Q) \tag{B-6}$$

where Q is now a constant delay.[8]

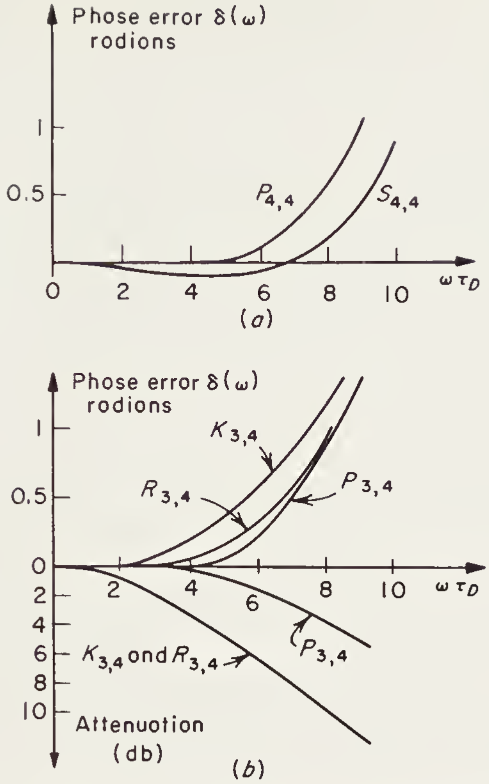


Fig. B-2*a*, *b*. Frequency response of all-pass delay networks (*a*), and low-pass delay net works (*b*). (*Data based on Ref.* 6.)

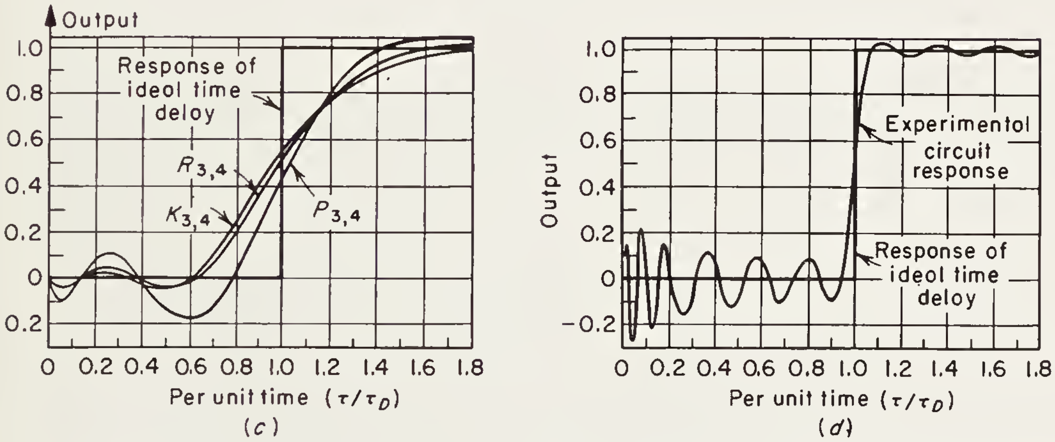


Fig. B-2*c*, *d*. Unit-step response of fourth-order low-pass delay networks (*c*), and of three cascaded identical low-pass networks of the type shown in Fig. B-1*a* (9 amplifiers, $n = 15$), (*d*). The step response of all-pass delay networks is practically unusable, since "precursor" oscillations start with the full step amplitude. (*Data based on Refs.* 5 *and* 6.)

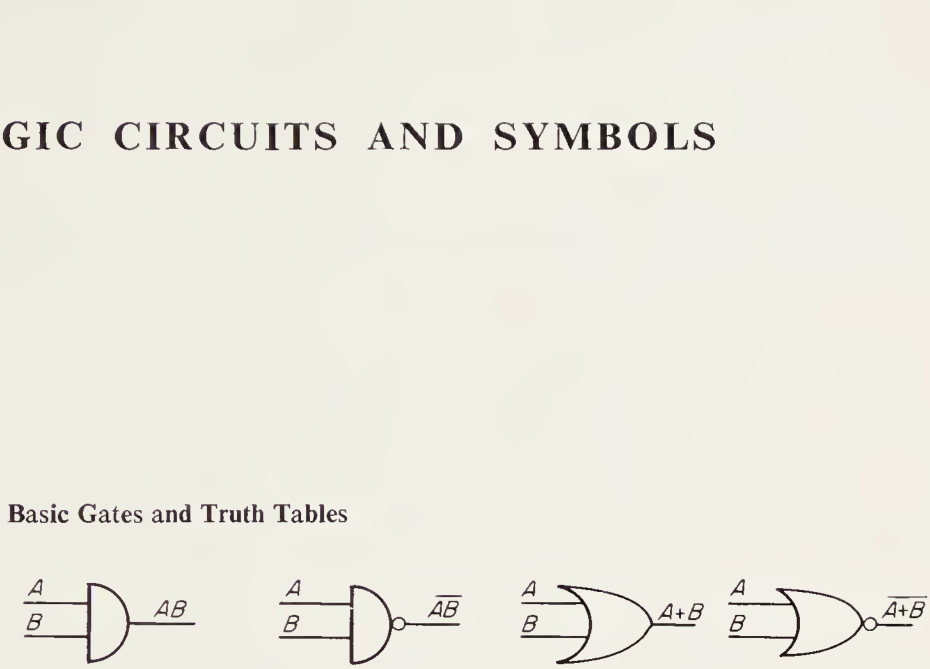
REFERENCES

1. Smith, G. W., and R. C. Wood: *Principles of Analog Computation*, McGraw-Hill, New York, 1959.
2. Rogers, A. E., and T. W. Connolly: *Analog Computers in Engineering Design*, McGraw-Hill, New York, 1960.
3. Jackson, A. S.: *Analog Computation*, McGraw-Hill, New York, 1960.
4. Johnson, C. L.: *Analog Computer Techniques*, 2d ed., McGraw-Hill, New York, 1963.
5. Single, C. H.: Time-delay Circuits for Analog Computers, *Tech. Bull.* 5059-1, Beckmann/Berkeley Div., Richmond, Calif., 1956; see also *Control Eng.*, October 1956.
6. King, W. J., and V. C. Rideout: Improved Transport-delay Circuits for Analog-computer Use, *Proc. 3d AICA Conf.*, Opatija, Yugoslavia, 1961, Presses Académiques Européennes, Brussels, 1962 (note bibliography).
7. Korn, G. A.: Network Approximations of Variable Time Delays Require Caution, *IRETEC*, February 1962.
8. Margolis, S. G., and J. J. O'Donnell: Rigorous Treatment of Variable Time Delays, *IEEETEC*, December 1963.
9. Vichnevetsky, R.: Analog-computer Simulation of a Time-dependent Delay Using the Concept of Generalized Transfer Function, *Ann. AICA*, April 1964.
10. Hansen, P. D.: New Approaches to the Design of Active Filters, *Simulation*, May and June 1966.
11. Nielsen, G.: Time-delay Approximations, Danish Technical University Lingby, 1968.
12. Gamp, J.: Génération de retards, *Report* EUR 240.f, EURATOM/CETIS, Ispra, Italy, 1963.
13. Hausner, A., and C. M. Furlani: Tables of Best Chebychev Approximants for Time-delay Simulation, TM-65-58, Harry Diamond Laboratories, Washington, D.C., 1965.
14. Hepner, C. F.: Improved Methods of Simulating Time Delays, *IEEETEC*, April 1965.

APPENDIX C

LOGIC CIRCUITS AND SYMBOLS

C-1. Basic Gates and Truth Tables



AND

$B \backslash A$	0	1
0	0	0
1	0	1

NAND

$B \backslash A$	0	1
0	1	1
1	1	0

OR

$B \backslash A$	0	1
0	0	1
1	1	1

NOR

$B \backslash A$	0	1
0	1	0
1	0	0

NOTE: In some types of logic, gate outputs can be ORed together. NAND or NOR gates also serve as *logic inverters* complementing a single input (1 becomes 0, and vice versa). Some gates have two complementary outputs, and some logic systems provide gates with inverting inputs.

C-2. Rules of Boolean Algebra

$$\left.\begin{aligned} A + B &= B + A \\ AB &= BA \end{aligned}\right\} \quad \text{(COMMUTATIVE LAWS)}$$

$$\left.\begin{aligned} A + (B + C) &= (A + B) + C \\ A(BC) &= (AB)C \end{aligned}\right\} \quad \text{(ASSOCIATIVE LAWS)}$$

$$\left.\begin{aligned} A(B + C) &= AB + AC \\ A + BC &= (A + B)(A + C) \end{aligned}\right\} \quad \text{(DISTRIBUTIVE LAWS)}$$

$$A + A = AA = A \qquad \text{(IDEMPOTENT PROPERTIES)}$$

$$A + B = B \text{ if and only if } AB = A \qquad \text{(CONSISTENCY PROPERTY)}$$

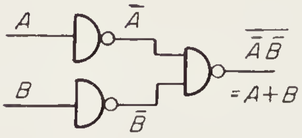
$$\left.\begin{aligned} &A + 0 = A \qquad A1 = A \\ &A0 = 0 \qquad A + 1 = 1 \\ &A(A + B) \equiv A + AB \equiv A \end{aligned}\right\} \quad \text{(LAWS OF ABSORPTION)}$$

$$\left.\begin{aligned} (\overline{A + B}) &\equiv \bar{A}\bar{B} \\ (\overline{AB}) &\equiv \bar{A} + \bar{B} \end{aligned}\right\} \quad \text{(DUALITY, OR DE MORGAN'S LAWS)}$$

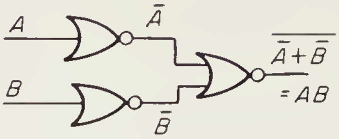
$$\bar{\bar{A}} \equiv A \qquad \bar{1} = 0 \qquad \bar{0} = 1$$

$$A + \bar{A}B \equiv A + B \qquad AB + AC + B\bar{C} \equiv AC + B\bar{C}$$

C-3. NAND/OR and NOR/AND Conversion (by de Morgan's Theorem)



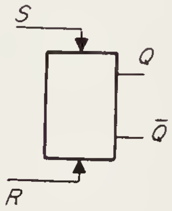
OR

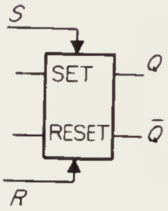


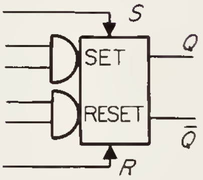
AND

All combinatorial logic can be implemented with NAND gates alone, or with NOR gates alone.

C-4. Flip-flops





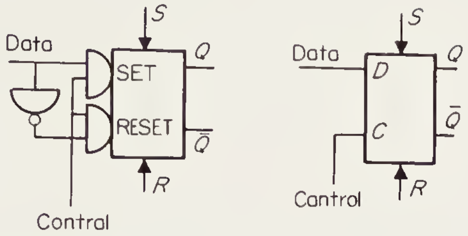


Reset/Set Flip-flop (RS Flip-flop). $R = 1$ (level or pulse) *resets (clears)* flip-flop: $Q = 0$, until $S = 1$ *sets* flip-flop ($Q = 1$). $R = S = 0$ leaves output *unchanged.* $R = S = 1$ is *illegal*, or acts like $R = 1$. Multiple set inputs or multiple reset inputs are ORed together.

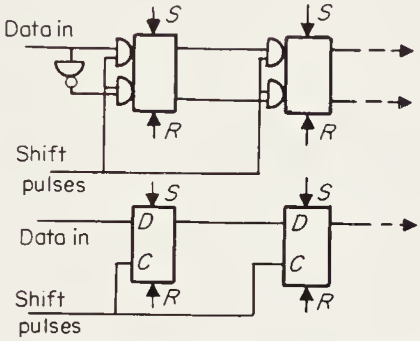
General-purpose Flip-flop with Enabling Gates. Many different types exist. Inputs may include inverting set and/or reset inputs, multiple gate inputs, etc. Frequently, the lower reset input is designed to override all other inputs.

Consult Manufacturers' Logic Manuals for exact logic, fan-out, logic-level tolerances and noise immunity, pulse duration and step rise time required to trigger flip-flops, etc.

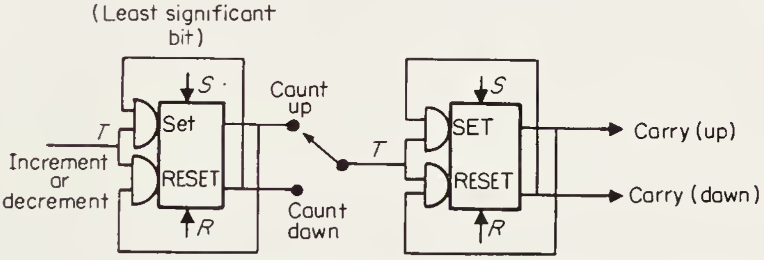
C-5. Applications of General-purpose Flip-flops



Data/Control (sometimes called Type D) Flip-flop. Output Q takes data-input value when control input goes to 1; acts as a binary sample-hold circuit. Important for jam transfer of data timed by control (strobe) pulses, and in shift registers.



Simple Shift Registers. Note possibility of parallel input through extra set and reset terminals.



Simple Binary Counter. Each counter flip-flop complements whenever its trigger (T) input changes to 1. Counter can be *preset* with S and R inputs.

AUTHOR INDEX

The author index refers, by page numbers, to entries in the *References and Bibliography* sections, which are arranged by subject matter at the end of each individual chapter.

Aaron, M. R., 210
Aasnaes, H. B., 211
Abe, R., 161
Adise, H., 253
Agarwal, V. K., 159
Aldrich, R. A., 300
Alliston, W. H., 255
Andelman, S. J., 299
Anderson, M. D., 360
Anderson, T. C., 258, 356
Andrew, A. M., 112
Andrews, J., 257
Angelo, E. I., 257
Ash, J. A., 210
Aus, H. M., 389

Bailey, S. J., 299
Baker, R. H., 210
Balakrishnan, A. V., 356
Barrack, C. M., 256
Barton, K., 209
Bauer, L., 360
Baum, R. V., 158, 159, 209, 257
Beadle, R., 162, 356
Bedient, C. K., 389
Bekey, G. A., 22, 41, 356, 387
Bell, B., 209
Bergersen, T. B., 210
Biernson, G. A., 253
Billinghurst, E. M., 110, 112
Bingulac, S., 355
Birt, D. R., 112
Bladowski, R., 112
Blatt, H., 211
Bloom, H. M., 160, 254
Bode, H. W., 257
Bodoia, M. J., 211
Bohn, E. V., 256
Bolie, V. W., 254
Borger, E. R., 299
Bowers, W. J., 299
Brennan, R. D., 389
Bright, R. L., 209
Brooks, S. H., 357
Broomal, J., 257
Brown, G. W., 355
Brown, T. S., 359
Brubaker, T. A., 111, 162, 210
Brunner, W., 162, 258, 356, 358
Brussolo, J. A., 110
Budzilovich, P. N., 299
Buntenbach, R. W., 299
Burdi, A. J., 299
Burgin, G. H., 298
Burr-Brown Research Corp., 111, 158
Byrd, J. S., 160

Cahn, L., 253
Caldwell, W. F., 162
Campbell, J. B., 300
Carroll, C. C., 159
Centner, R. M., 211
Chan, S. K., 359
Chance, B., 255, 256
Chapelle, W. E., 388
Clark, W. R., 111
Cleverley, D. S., 159
Clymer, A. B., 358
Cochran, D. S., 159
Cohen, E., 256
Cohen, J. M., 211
Comley, W., 160, 254
Conant, B. K., 112, 159, 162, 210, 298
Connelly, M. E., 254, 389
Connolly, T. W., 23, 404
Coombs, D. J., 300
Corbett, J., 356
Crawford, J. I., 211
Crayford, R. M., 256
Cullin, A., 210

Dastur, S. P., 356
Davis, H. B., 255
Davis, W. R., 111
De Angelis, D. P., 111
De Backer, W., 298, 360
Deeley, E. M., 258
Deering, C. S., 112
Deiters, R. M., 254
DeLotto, I., 211
Demrow, R. I., 113
Deveraux, H. R., 210
Dike, L. L., 389
Dobbin, R., 112
Dodds, W. R., 389
Doganovskii, I., 160
Dole, F. E., 110, 253
Dow, P. C., 110, 111
Dujmovič, M., 161
Dunsmore, C. L., 359
Dyntschaever, D., 388

Eckes, H. R., 159, 210, 297, 359
Ehlers, H. L., 112, 209, 253, 298
Eide, M. A., 388
Electronic Associates, Inc., 258, 359
Evans, A. D., 210
Eychhoff, P., 358

Fannin, B. B., 359
Fasal, J., 210
Favreau, R. R., 357
Feldbaum, A. A., 356
Fifer, S., 22
Fineberg, M. S., 389
Fink, H. J., 358
Finlay, D. J., 209
Fisher, M. E., 22, 23, 254, 358, 359
Fitsner, L. N., 160, 254, 257
Flockhart, D. G., 256
Fogarty, L. E., 355, 358
Foote, D. D., 159
Forrester, J. H., 388
Franklin, M. A., 389
Franks, R., 357
Free, M. G., 255
Freeman, J., 210
Freyling, N., 111
Fritz, R., 160, 254
Fuchs, H., 256
Fulcher, E. M., 300
Furlani, C. M., 404

Gafni, G., 255
Gamp, J., 404
Gans, F., 113
Garner, K. C., 161
Geffe, P. R., 22
Gelman, R., 387
Gerard, R. C., 389
Gerlach, A. A., 161
Giacoletto, L. J., 113
Gibbons, J. F., 160
Gibson, J. E., 253
Gilbert, B., 255
Gilbert, C. P., 22
Gilbert, E. G., 258, 300, 356, 357, 387
Gilbert, E. O., 253
Gilbert, J., 110, 253
Giles, J. N., 111
Giloi, W., 22, 160, 211, 254, 297, 358, 388, 389
Ginn, D., 299
Ginzburg, S. A., 162, 258, 388
Giser, S., 258
Gleghorn, P., 257
Goldberg, E. A., 111, 255, 257, 355
Goldhirsh, I. L., 159, 390
Goltz, J. R., 41, 211
Gonzalez, R. S., 355, 357
Graber, G. F., 360
Gracon, T. J., 159, 298, 390
Grado, G. R., 258
Graeme, J. G., 113
Graves, W. L., 389
Gray, H. J., 209
Gray, P. E., 158, 209
Green, C., 298
Greenwood, D. T., 356, 358
Greenwood, I. A., 253, 257
Greiner, R. A., 258
Gringle, B., 256
Gulick, J. F., 209
Gundlach, F. W., 257
Gunning, W. F., 253
Gupta, S. C., 360
Gurin, L. S., 357
Gurski, R. J., 160
Gutz, S., 389
Guyton, R. D., 111

Haas, D. L., 299
Haase, W. C., 255
Halbert, P. W., 358
Hamer, H., 112, 298
Hancock, E. J., 161
Handler, H., 112
Hannauer, G., 298
Hansen, P. D., 404
Hara, H. H., 358
Hardcastle, I., 210
Harris, H. E., 161
Harris, M., 253
Harrison, T. J., 211
Hartmann, J., 255, 258

Hausner, A., 22, 23, 159, 404
Heim, E., 111, 160, 258
Hepner, C. F., 404
Hertog, J. M., 159
Hilbinger, A. R., 256
Hirschfeld, R. A., 112
Hoag, D. M., 160
Hochwald, W., 112, 209, 253, 298
Hoeschele, D. F., 209
Hoffait, A. H., 111
Hofman, L. G., 358
Hofstadter, R., 253
Holbrook, J. G., 113
Holdam, J. V., 253, 257
Horn, H. S., 160
Horwitz, R. D., 298
Howe, R. M., 158, 159, 255, 258, 298, 355, 356, 358, 359
Huelsman, L. P., 22, 113
Hunt, J. M., 256
Huskey, H. D., 22, 41, 110, 112, 209, 253, 258, 297, 298, 356, 358–360

Ignatyev, M. B., 355
Ingalls, V. W., 388
Ingerson, W. E., 112
Iwata, J., 358, 387, 388

Jackson, A. S., 22, 404
Jacoby, E. A., 389
Janac, K., 357
Jarrett, R. J., 388
Johnson, C. L., 22, 404
Johnston, S. A., 110
Jones, J. P., 299
Jonsson, J. J., 298

Kahne, S. J., 355
Kamm, L. J., 298
Karnopp, D. C., 357
Karplus, W. J., 22, 23, 41, 110, 356–358, 388, 389
Kataoka, S., 256
Kavanaugh, W. P., 357
Kay, B. G., 211
Keay, C. S., 210
Keene, D. H., 298
Keister, G. L., 256
Keller, R. E., 356
Kelley, T., 161
Kemp, N. H., 388
Kennedy, J. D., 298
Kennewell, J. A., 210
Kerwin, W. J., 22
Kettel, E., 257
Kindle, W. K., 258
King, T. A., 110
King, W. J., 404
Kingma, Y. J., 210
Kislitsyn, S. G., 161
Klein, G., 159
Klerer, M., 299, 357
Kley, A., 22, 112, 159, 160, 210, 258
Klipec, B., 299
Koerner, H. E., 112, 159
Kogan, B. Y., 22, 111
Kokotovíc, P., 355
Korn, G. A., 22, 23, 41, 110–112, 159, 161, 162, 209–211, 253, 255–258, 297–300, 355–359, 387–389, 404
Korn, T. M., 257, 258, 359, 360
Kosako, H., 358
Kovach, L. D., 160, 254, 356
Kovatch, G., 256
Kruper, A. P., 209
Kuchlewski, T. R., 299
Kurokawa, K., 254, 298
Kusto, T. J., 297
Kvamme, E. F., 209

Langill, A. W., 359
Larrowe, V., 161
Lauber, R., 22
Lazarus, H., 254
Leite, C., 159
Levenstein, H., 254
Levin, M. H., 113
Levine, L., 23
Liban, E., 358
Liebert, T. A., 41
Lilamand, M., 257
Linebarger, R. N., 389
Litvin, F. L., 161
Lofgren, L., 255, 387
Loveman, B. D., 110, 253, 297, 298
Lovering, W. P., 22
Lyden, A., 209
Lyubarskii, Y. Y., 388

McCoy, R. D., 161, 254
McCullough, W., 299
MacDonald, R. A., 389
McDonald, R. K., 112
McGhee, R. B., 356
McGrath, R. J., 358
MacKay, D. M., 23, 254, 358, 359
McLeod, J., 23, 359
MacNee, A. B., 111, 257
MacRae, D., 253, 257
Manfredi, P. F., 211
Marchant, H., 161
Margolis, S. G., 404
Marimon, R. L., 299

Marjanovič, S., 211
Markson, E. E., 258
Marsh, J. M., 299
Marshall, B. O., 355
Marsocci, V. A., 111
Marston, G. P., 299
Maslov, A. A., 160, 254, 255, 257
Matsuno, S., 355
Matyas, J., 357
Maybach, R. L., 255, 258, 356, 357
Mayer, R. E., 160
Medkeff, R. J., 159
Meissinger, H. F., 110, 160, 161, 254, 355, 356
Mengel, A. S., 253
Meserve, W. E., 256
Meyer, R. A., 255
Meyer-Brötz, G., 111, 112, 159, 209, 297
Mickey, M. R., 357
Middlebrook, R. D., 113
Milicevič, D., 257
Miller, A., 299
Miller, K. S., 110, 355
Miller, W., 113
Millman, J., 158, 209, 257
Mitchell, B. A., 209, 357
Mitchell, E. E., 258, 388
Mitra, S. K., 210
Miura, T., 110, 111, 160, 161, 257, 358, 387, 388
Molcho, J., 255
Monastyrshin, G. I., 359
Moran, R. A., 300
Morrill, C. D., 158, 159, 209, 254, 257, 258
Morrison, R., 113, 299
Morton, R. R., 255
Moschytz, G. S., 22
Moser, J. H., 357
Moss, R. Y., 113
Mott, H., 159
Moulton, A. B., 159
Mrazek, D., 210
Munson, J. K., 357
Murray, F. J., 110, 355
Muterspaugh, M., 355
Mynall, D. J., 160

Nagata, M., 111
Naka, M., 388
Nathan, A., 254, 255
Naylor, J. R., 112
Nelson, K. F., 359
Netter, Z., 254
Nettleton, L. A., 110, 253
Neustadt, L. W., 355–357
Newman, D. J., 358
Nielsen, G., 404
Nilsen, R. N., 388
Noaks, D. R., 211

O'Donnell, J. J., 404
Offner, F. F., 112
O'Grady, E. P., 355, 388
Oliver, B. M., 159
Oxeni, J., 256
Ozdes, D., 112

Palazzini, N., 211
Palumbo, M., 111
Panico, J. J., 159
Parent, R. J., 159
Parezanovič, N., 161
Paterson, W. L., 160
Paul, R. J. A., 256
Pawluk, H., 159
Paynter, H. M., 162, 359
Pearman, C. R., 210
Pederson, C. N., 161
Pelle, J., 389
Peretz, R., 161
Petrovic, R., 256
Petternella, M., 161
Pfeiffer, P. E., 158, 255
Philbrick-Nexus Research (Teledyne), 111, 158, 161, 255
Phillips, A. B., 158, 209
Pierce, D. E., 299
Pinckney, R. P., 258
Polimerou, L. G., 161
Polonikov, D. E., 112
Popodi, A. E., 210
Post, P. B., 162
Pracht, C. P., 210, 299
Presson, A. G., 358
Price, R., 256
Puckett, T. H., 209
Purlov, Y. G., 160
Pyne, I. B., 355

Raamot, J., 211
Raby, J. S., 360
Rajaraman, V., 358
Rastrigin, L. A., 357
Raymond, F. H., 258
RCA Laboratories, 255
Ream, N., 254
Reed, E. F., 389
Reed Company, 357
Richards, J. C., 389
Rideout, V. C., 358, 404
Riebman, L., 257
Rigas, H. B., 300

Ritchie, C. C., 254
Roberge, J. K., 210
Rogers, A. E., 23, 404
Rogers, S., 297
Rook, P. E., 298
Rosenkranius, L., 161
Ruberti, A., 161
Rubin, A. I., 22, 356, 357, 360, 389
Rubinoff, M., 209
Rubinshtein, J. S., 357
Rudin, M. B., 211
Rudnick, S., 110
Rybashov, M. V., 356

Sack, H. S., 257
Saenz, R. G., 300
Sage, M. W., 298
Sargent, R. S., 299
Savet, P. H., 255
Savitt, A. 211
Scanga, W. A., 256
Schaefer, R. A., 256
Schick, L. L., 211
Schmid, H., 159, 209, 211, 255–258, 388
Schmidt, W., 257, 299
Schneider, W., 160, 209, 257
Schreier, D., 160, 254
Schumer, M. A., 357
Schwartz, J., 112
Scott, V. P., 299
Seay, P. A., 254
Sebestyen, G., 209
Seegmiller, W. R., 256
Seehuns, T. K., 389
Sellars, H. L., 357
Serlin, O., 389
Sheehan, C. J., 111, 355
Sheingold, D. H., 113
Shen, C. N., 359
Sherchant, I., 161
Siljac, D., 256
Single, C. H., 110, 111, 113, 404
Skramstad, H. K., 356, 388
Sleehan, C. J., 258
Smith, B. R., 209
Smith, D. A., 355
Smith, G. W., 160, 404
Smith, L., 113
Smith, P. D., 110
Smolov, V. B., 161
Solomon, J. E., 111
Solomonoff, R., 112
Soltes, A. S., 254
Soma, G. N., 389
Sommer, H., 211
Sommerville, M. J., 256
Sosenskii, N. L., 111
Soudack, A. C., 355
Soxman, E., 255
Spencer, M., 161
Spirer, H. F., 110
Starr, D. A., 298
Stata, R., 111, 112, 162
Steiglitz, K., 357
Stein, M. L., 298
Steingroewer, H., 359
Stern, T. E., 161
Sternberg, S., 257
Sternick, L., 254
Stewart, E. C., 357
Stewart, E. L., 299
Stifle, J., 211
Stojanovič, B., 257
Strauss, J. C., 159, 298, 358, 389, 390
Stricker, J. L., 258
Sunstein, D. E., 160
Sykes, R. P., 110, 298

Talkin, A. I., 356
Tarpley, R. E., 111
Taub, H., 158, 209, 257
Terasaki, R. M., 359
Terrazas, C., 359
Thaulow, P., 258
Thedchanamoorty, N., 360
Thompson, M. D., 389
Thornton, R. D., 111
Tobey, G. E., 113
Tomalesky, A. W., 359
Tomovič, R., 23, 110, 355, 359, 387, 388
Tompkins, J., 209
Tossman, B. E., 389
Truxal, J. G., 253
Turner, R. M., 356
Tutelman, D. M., 388
Tuteur, F. B., 253

Vance, P. R., 299
Van Steenkiste, G. J. V., 159, 338
Van Wauwe, A., 298
Vichnevetsky, R., 258, 357, 358, 387, 388, 404
Vidal, J., 388
Visman, A. C. P., 161

Waha, J. P., 357
Wait, J. V., 387, 388
Waldhauer, F. D., 112, 211
Waldschmidt, K., 389
Walker, M., 211
Walker, R. E., 160
Walker, R. M., 300
Walters, L. G., 257

Wang, H., 359
Welling, B., 112
Wentzel, V., 161
Whigham, R. H., 160, 254
White, R. C., 357
Widlar, R. J., 111, 162, 210
Wilde, D. J., 356
Wilkins, J., 299
Wilkinson, J. R., 211
Wilkinson, R. H., 162
Willard, D. A., 389
Williams, A. J., 111
Williams, R. W., 161
Wilmshurst, T. H., 113
Wilson, B. R., 389
Wilson, J. P., 256
Wingrove, R. C., 360
Winkler, H., 160, 254
Witsenhausen, H., 159
Wollensen, D. L., 209
Wong, M. S., 359
Wood, R. C., 160, 404

Yagi, A., 254, 298
Young, R. W., 254

Zampino, J., 359
Zenner, R. E., 161
Zusi, F. C., 299

SUBJECT INDEX

References in the subject index are to section numbers, not page numbers. Note that section numbers are displayed at the top of each page for convenient reference.

Absolute-value circuit, **4**-10, **5**-14
Active filter, **1**-10
"Active" flip-flop, **7**-11
Active switching circuit, **5**-5
ACTRAN, **7**-4*c*
ADC (*see* Analog-to-digital converter)
Adjoint equation, **8**-16*a, b*
Algebraic equations, **8**-8
Aliasing, **9**-2
Amplitude modulation, **4**-12
Amplitude selector, **4**-3, **4**-10, **4**-11
Analog comparator (*see* Comparator)
Analog memory, **2**-2*c*
Analog-to-digital converter, **5**-10, **5**-11, **5**-15
  cascade, **5**-14
  continuous, **5**-13*b*
  dual-slope, **5**-12
  errors in, **5**-15
  integrating, **5**-12
  interface, **7**-8, **7**-10
  parallel, **5**-14
  ramp/comparator, **5**-12, **5**-13*a*
  subranging, **5**-15
  successive-approximation, **5**-13*c*
APACHE, **7**-4*c*
APE II, **7**-2*a*
Aperture time, **5**-17
Applied Dynamics, Inc., **7**-2, **7**-4*c*
APSE, **7**-4*c*
ARM terminal (*see* Potentiometer)
Assembly language, **9**-15
Astable multivibrator, **4**-12
Automatic data channel, **7**-12*b*
Automatic patching, **7**-4*c*
Automatic scaling, **4**-10

Back resistance, **5**-1*a*
Backlash, **8**-18
Balancing amplifiers, **3**-37, **7**-6*a*
Bang-bang control, **8**-16*b*
BASIC, **9**-12
BCD (binary-coded decimal) number, **5**-7
Bias-current offset, **3**-29, **3**-32, **3**-33
Binary-coded decimal number, **5**-7
Bistable operational-amplifier circuit, **4**-12
Block transfer, **7**-12*b*
Bode plot, **7**-34
Boolean algebra, Appendix C
Bootstrap unloading circuit, **1**-10
Boundary minimum, **8**-7*b*
Boundary-value problem, **8**-13, **8**-16
Breakpoint, **4**-2, **4**-4, **4**-15
  (*See also* Function generation)
Breakpoint drift, **4**-2
Bridge, switch, **5**-2
Bus gate, **7**-8, **7**-10

Calibration, **3**-37, **6**-15, **7**-6*a*
Canonical equations, **8**-16*a, b*
Capacitor:
  measurement, **3**-9
  oven, **3**-9
  specifications and errors, **3**-8, **3**-20
Capcoder, **5**-7
Chopper stabilization, **3**-36
Circle test, **3**-24
Clear-and-strobe transfer, **7**-9*c*
Clock, analog-computer, **7**-3*b*
Clock interrupt, **9**-7
CMRR (*see* Common-mode rejection)
Coefficient of restitution, **8**-18
Coefficient setting (*see* Digital-coefficient unit; Potentiometer)
Combined analog-digital simulation, **2**-5
  errors, **9**-2 to **9**-6
  programming, **9**-13*c*

Combined simulation, **9**-13
  errors, **9**-3, **9**-4
Common-mode gain, **3**-26
Common-mode rejection, **3**-26, **3**-33
Comparator, **4**-9
  hysteresis, **4**-9*b*
  integrated-circuit, **4**-9
Compensation of phase shift:
  integrator, **3**-20
  potentiometer, **3**-6
Component accuracy, **3**-2
COMPUTE mode, **1**-2, **7**-3*a*
  (*See also* Mode control)
Computed gradient, **8**-14
Concentrator, **7**-4*c*
Constraints, **8**-7, **8**-15, **8**-16
  artificial, **8**-9*b*
  dynamical, **8**-9*a*
  redundant, **8**-9*b*
  (*See also* Linear programming)
Control register, **7**-8
Control variable, binary, **2**-2
Conversion rate, **5**-10 to **5**-15
Conversion time, **5**-10 to **5**-15
Coordinate transformation, **6**-17
Coulomb friction, **8**-19
Creeping random search, **8**-14, **8**-15
Criterion function (*see* Optimization)
Crossbar matrix, **7**-4*c*
Crosstalk, **7**-4
Current-address counter, **7**-12*b*
Current offset:
  bias, **3**-29, **3**-32, **3**-33
  differential, **3**-29, **3**-32, **3**-33
  (*See also* Offset)
Current source, **1**-10
  switched, **5**-8
Current switching, **5**-8
Curve follower, **7**-5*b*
Cycle stealing, **7**-12

DAC (*see* Digital-to-analog converter)
Data channel, **7**-12*b*
Data-reconstruction errors, **9**-2, **9**-3, **9**-5, **9**-6
Data transfer, **7**-9*c*
D-c amplifier (*see* Operational amplifier)
DCU (digital coefficient unit), **7**-4*b*
Deadspace characteristic, **4**-10, **8**-18
Decoupling circuit, **4**-9*b*, **7**-14
Delay, digital processing (*see* Digital processing delay)
Delay operator, **2**-3, Appendix B
  (*See also* Function storage)
Demodulation, **5**-1
  (*See also* Chopper stabilization)
Depletion mode, **5**-4*b*
Device selection, **7**-9*b*
Diagnostic program, **9**-16
Dielectric absorption, **3**-9, **3**-20
Difference equations, **2**-3
Differential amplifier, **3**-25 to **3**-30, **6**-6
Differential analyzer, **1**-2, **1**-4, **1**-8
Differential gain, **3**-26
Differentiating circuit, **1**-10
Digital attentuator, **7**-4*b*
Digital coefficient unit, **7**-4*b*
Digital computer in hybrid system, **7**-8
Digital logic, patched, **2**-2, **7**-3*c*
Digital processing delay, **2**-5, **9**-2, **9**-4 to **9**-6
  errors due to, **9**-4, **9**-5
Digital-to-analog converter, **5**-6
  double-buffered, **7**-9*c*
  fixed-reference, **5**-6
  interface, **7**-8
  multiplying, **5**-6
Digital voltmeter (*see* Analog-to-digital converter)
Diode bridge, **5**-2
Diode characteristics, **4**-2
Diode limiter (*see* Limiter, diode)
Diode switches, **5**-2
Direct memory access, **7**-8, **7**-12, **9**-7, **9**-13*d*
Dissipation coefficient, **3**-9*b*
Division, **6**-11*b*, **6**-16
  with automatic rescaling, **8**-6
  how to avoid, **1**-11
  steepest descent, **8**-9*b*
DMA (*see* Direct memory access)
Dow's capacitor model, **3**-9*b*
Drift, **3**-32, **3**-33, **3**-36
  of track-hold circuit, **5**-17
  (*See also* Offset)
Dry-disk clutch, simulation, **8**-20
Dry friction, **8**-19

ECL (emitter-coupled logic), **7**-3, **7**-14
Educational computer, **7**-2*a*
Eigenvalue problem, **8**-13
Elastic stop, **8**-18
Electronic Associates, Inc., **1**-2
Emitter-coupled logic, **7**-3, **7**-14
Enable flip-flop, **7**-12*a*
Enhancement mode, **5**-4*b*
Equation error, **8**-12
Error estimation, **8**-4, **8**-5
Errors:
  in digital-to-analog conversion, **5**-9
  in hybrid computation, **9**-2 to **9**-6
  in solution of linear differential equations, **3**-22 to **3**-24
  in track-hold circuit, **5**-17
Euler-Lagrange equations, **8**-16*a*
Executive program, **9**-11

Expander, **7**-4*c*
Extrapolation, **9**-3
Extreme-value holding circuit, **4**-11

Feedback ratio, **3**-11
Feedforward circuits, **3**-35
FET (*see* Field-effect transistor)
Field-effect transistor:
  in amplifier, **3**-33
  as chopper, **3**-36
  as switch, **5**-4
Film resistors, **3**-7*b*, **5**-7
First-order hold, **9**-3
Flag, **7**-10, **7**-11
Floating control circuit, **5**-1*e*
Flow chart, **2**-2
Folding error, **9**-2
Follower amplifier, **3**-30, **5**-17
FORTRAN, **9**-11 to **9**-15
  hybrid, **9**-13
Forward resistance, **5**-1*a*, **5**-17
Four-diode bridge, **4**-6
Frame time, **9**-4, **9**-13
Frequency modulation, **4**-12
Frequency response:
  large-signal, **3**-2*a*
  small-signal, **3**-2*a*
Friction, simulation of, **8**-19
Friction clutch, **8**-20
Function generator:
  cathode-ray-tube, **4**-17*c*
  diode, **4**-14, **4**-15, **4**-19
  diode squaring network, **6**-2, **6**-3
  hybrid methods, **9**-8 to **9**-10
  logarithmic, **4**-16
  signal generator, **4**-12
  sine/cosine, **6**-20
  two or more variables, **4**-18, **9**-9
  varistor, **4**-17*b*
Function storage, **9**-7
  (*See also* Time-delay simulation)
Functional, **8**-13

Gain error, **9**-2
Generalized integration, **1**-13, **6**-24, **8**-9*b*
Glitch, **5**-9
Gradient method, **8**-14, **8**-16
  (*See also* Steepest descent)
Granularity, simulation of, **8**-20
Ground loop, **7**-13
Ground system, **7**-13, **7**-14

Half-scale accuracy, **3**-1
Hamiltonian, **8**-16*a*, *b*
HELIPOT Corporation, **3**-4
HI terminal (*see* Potentiometer)
HOLD mode, **1**-2, **2**-2, **5**-17
  (*See also* Mode control)
Hybrid code, **9**-5*e*
Hybrid executive, **9**-14
Hysteresis:
  bistable circuit, **4**-12
  of comparator, **4**-9*b*
  simulation of, **8**-20

Identification (*see* Model matching)
Implicit computation of sine and cosine, **6**-24
Impulse response, **1**-9
Induction resolver, **4**-13
Industrial ground, **7**-13
Inelastic stop, **8**-18
Inequality constraints (*see* Constraints; Linear programming)
Initial conditions, **1**-5
  (*See also* RESET mode)
INITIAL-RESET mode, **2**-2*d*
Input impedance of operational amplifier, **3**-14, **3**-30
Input/output bus, **7**-9
Integrated circuit:
  amplifier, **3**-33, **3**-35
  DAC, **5**-8
  multiplier, **6**-6
  switch, **5**-4
Integrator, **1**-2, **1**-5, **3**-20, **3**-21
  error analysis, **3**-20, **3**-21
  noninverting, **3**-28
  as track-hold circuit, **5**-17
  (*See also* Mode control)
Interface, **7**-8 to **7**-12
  requirements, **7**-8
Interlace, **7**-12
  (*See also* Direct memory access)
Intermittent-HOLD, **9**-5*b*
Interpolation:
  in function generation, **4**-18, **9**-8
  in hold circuits, **9**-3
  (*See also* Function generator)
Interpolative hold, **9**-3
Interpreter, **9**-12
Interrupt, **7**-8, **7**-11, **9**-7, **9**-12, **9**-13
Inverse function, **4**-14, **8**-1
Inverse resolution, **6**-18, **6**-22
Inverted transistor, **5**-3*b*
I/O (input/output) bus, **7**-9
Isolation transformer, **7**-13
Iterative analog computation, **2**-1 to **2**-5
  with digital control, **9**-13*b*
Iterative differential analyzer, **8**-14
Iterative optimization (*see* Optimization)

Jam transfer, **7**-9*c*

Junction field-effect transistor as a switch, **5**-4*a, c*

Kokotovic system, **8**-14*c*

Ladder network, **5**-7, **5**-8
Lagrange multiplier, **8**-16
  (*See also* Constraints)
Leakage:
  capacitor, **3**-9
  diode, **4**-2
  FET, **3**-33
  patchbay, **7**-4
  transistor, **3**-33, **5**-3
Least-significant bit, **5**-6, **5**-9
Library routine, **9**-12
Limit stop, **8**-18
Limiter:
  diode: bridge, **4**-6, **4**-10
    feedback, **4**-3
    series, **4**-3
    series-shunt, **4**-3, **6**-3
    shunt, **4**-3
  high-frequency operation, **4**-5
    precision, **4**-10
    transistor, **4**-8
    Zener-diode, **4**-3*c*
Linear programming, **8**-10
Linearization, **8**-2
Linkage (*see* Interface)
Linking loader, **9**-11
Loader, **9**-11
LOCUST, **7**-2, **9**-15
Logarithm-function generator, **4**-16
Logic, patched, **7**-3
Logic circuits, **7**-3, Appendix C
Loop gain, **3**-11
LSB (least-significant bit), **5**-6, **5**-9

Machine variable, **1**-2
Maintenance, **7**-7, **9**-16
Maxima and minima, **8**-7
Maximum-holding circuit, **4**-11
MDAC (*see* Digital-to-analog converter, multiplying)
Measured-gradient method, **8**-14*d*, **8**-16*c*
Measurements, operational-amplifier specifications, **3**-37, **3**-38
MEI (minimum-excursion indicator), **7**-6*b*
Memory pair (*see* Track-hold pair)
Metal-oxide-semiconductor field-effect transistor (MOSFET), **3**-36, **5**-4, **5**-16
Miller effect, **3**-19
Minimum-excursion indicator, **7**-6*b*
Minimum-holding circuit, **4**-11
Mode control, **1**-2, **5**-16, **5**-17, **7**-3
Model matching, **8**-12, **8**-13
Modulated waveforms, generation, **4**-12
Modulation, **4**-12, **5**-1
  (*See also* Chopper stabilization)
Monitor program, **9**-14
Monostable multivibrator:
  operational-amplifier circuit, **4**-12
  use of, in patched logic, **7**-3*c*
Monte Carlo studies, **2**-5
MOSFET switch, **3**-36, **5**-4, **5**-16
Most-significant bit, **5**-6
MSB (most significant bit), **5**-6
Multicomparator, **4**-10
Multiplexer addressing, **7**-12*b*, **9**-13
Multiplier, **6**-1
  AM/FM, **6**-8*d*
  analog-digital, **6**-8*a*, **6**-14
  cathode-ray-tube, **6**-1
  diode, **6**-2
  for division, **6**-15
  feedback-type, **6**-7, **6**-8
  four-quadrant, **6**-5
  Hall effect, **6**-8*c*
  heat-transfer, **6**-8*b*
  logarithmic, **4**-16
  magnetoresistance, **6**-8*c*
  mark-space, **6**-9 to **6**-13
  modulation-type, **6**-8*d*
  photo-resistor, **6**-8*b*
  pulsed-attenuator, **6**-9 to **6**-13
  quarter-square, **6**-2
  servo, **6**-1, **6**-8
  slave, **6**-7
  test circuits, **6**-15
  time-division, **6**-9 to **6**-13
  transconductance, **6**-6
  triangle averaging, **6**-4
  two-channel, **6**-14
Multivibrator circuit, **4**-12

Noise, **3**-15, **3**-21, **3**-29
  in computer system, **7**-13
  digital-system, **7**-14
  in operational amplifier, **3**-15, **3**-29
Nominal solution, **8**-3
Noninverting operational amplifier, **3**-25 to **3**-30
Nonsaturating switch (*see* Current switching; Switch, active)
Nyquist-Bode law, **3**-34

Offset:
  d-c amplifier, effect of, **3**-15, **3**-29
  switch, **5**-3, **5**-4*d*
Offset error in hybrid computation, **9**-2
One's-complement code, **5**-6

OPERATE mode (*see* COMPUTE mode)
Operating system, 9-14
Operational amplifier, **1**-10
  differential and noninverting, **3**-25 to **3**-30
  error analysis, **3**-10 to **3**-17
  measurements, **3**-37, **3**-38
  specifications, **3**-31 to **3**-36
  stability, **3**-34
Optimization, **8**-7
  functional, **8**-12, **8**-16
  iterative, **8**-14
  parameter, **8**-13
Optimum-gradient method, **8**-14
Oscillator, **4**-12, **4**-13
  smoothing, **4**-15, **6**-3
Oscilloscope display, **7**-5
Overload:
  in hybrid computation, **9**-13*d*
  in optimization problem, **8**-15
Overload indication, **7**-6*b*

PACE (*see* Electronic Associates, Inc.)
Padé approximation, Appendix B-2
Parameter estimation (*see* Model matching)
Parameter-influence coefficient (*see* Sensitivity)
Parameter optimization (*see* Optimization)
Parameter perturbation, **8**-11
Parameter sensitivity, **8**-4, **8**-14*c*
Parameter tracking, **8**-12
Partial differential equations, **8**-17
Partial fractions, **2**-4
Partial system test, **2**-1, **9**-6
Party-line I/O bus, **7**-9
Patched logic, **7**-3
Patching, automatic, **2**-5, **7**-4*c*
Patching systems, **7**-4
Paterson transdiode, **4**-16
Pedestal, **5**-1*e*
Perturbation:
  of parameter, **8**-11
  random, **8**-15
Perturbation method, **8**-1 to **8**-4, *9-5e,* **9**-6
Phase compensation, summing amplifier, **3**-19
Phase errors, **3**-2*b*
  effect of, on solution of linear differential equations, **3**-22 to **3**-24
Phase inverter, **1**-2
  errors, **3**-19
Phase margin, **3**-34
  measurement, **3**-38
Phase-shift oscillator, **4**-13
Phase shifter, **1**-10, **3**-28
Photoformer, **4**-17*c*
Piecewise-linear circuit analysis, **4**-2, **4**-4
Polar coordinates, **6**-17
POLAR resolver mode, **6**-22
Polynomial hold, **9**-3
Pontryagin's maximum principle, **8**-16*a, b*
Potentiometer:
  coefficient-setting, **1**-3, **3**-4 to **3**-6, **7**-4*b*
  design, **3**-4
  dynamic errors, **3**-6
  granularity, **8**-20
  loading, **3**-5
  three-terminal, **3**-5
POTSET mode, **7**-3
Power supplies, **7**-2*c*
Precision limiter, **4**-10, **5**-14, **8**-18
Precompiler, 9-15
Prediction in hybrid computation, **9**-3 to **9**-5
Priority for direct memory access, **7**-12
Priority interrupt, **7**-11
  (*See also* Interrupt)
Problem check, **7**-6*c*
Processing delay, digital (*see* Digital processing delay)
Program check, **7**-6
Program interrupt, **7**-11
Programmed input/output, **7**-9
Pushbutton logic, **7**-3*a*

Quadrant switching, **6**-21
Quantization error, 9-2
Quantizing circuit, **4**-10, **8**-20

Rate limit, **3**-2*c*
Rate resolver, **6**-21*b,* **8**-9*b*
Readout system, **7**-5
Real-time executive, 9-14
Real-time FORTRAN, **9**-13*e*
Recorder, **7**-5
Reentrancy, **9**-13*e*
Reeves Instrument Co., **4**-15
Reference power supply, **1**-3
Relay, simulation of, **8**-20
REP-OP (*see* Repetitive analog computer)
Repetitive analog computer, **2**-2, **6**-23
Request flip-flop, **7**-11
RESET mode, **1**-3, **2**-2, **5**-17
  (*See also* Mode control)
RESET pulse, **2**-2
  (*See also* Mode control)
Resistors, **3**-7
Resolution of digital-to-analog converter, **5**-9

Resolver, **6**-17
  electronic, **6**-20 to **6**-24, **8**-9*b*
  induction, **6**-19
  servo, **6**-17, **6**-19
Restitution coefficient, **8**-18
Resubstitution check, **8**-5
Reverse leakage (*see* Leakage, diode)
Ripple filter, **4**-15, **6**-12, **6**-16
Rolloff network, **3**-34

Sample-hold circuit (*see* Track-hold circuit)
Sampled-data operations, **2**-3
Sampling, **9**-3, **9**-4
Sampling circuits:
  for multiplication, **6**-23
  for sine generation, **6**-23
Sampling pulse, **2**-2
Sampling rate, **9**-3
Satellite trajectory, **8**-3
Saturation, simulation of, **8**-18
Saturation voltage, transistor, **5**-3
Scaling, **1**-2, **1**-6
  automatic, **8**-6
Semielastic stop, **8**-18
Sense line, **7**-8, **7**-10, **9**-13*d*
Sensitivity, **1**-10
  (*See also* Parameter sensitivity)
Service report, **7**-7
Servo recorder, **7**-5*b*
Servomultiplier, **6**-1
Setting stability, **3**-4
Settling time, **5**-9
Shielding of patchbay, **7**-4
Short-circuit transfer impedance (*see* Transfer impedance)
Sign bit, **5**-6
Sign multiplier, **4**-9, **8**-9*b*
Signal-generator circuits, **4**-12
Simulation, **1**-1
  combined analog-digital (*see* Combined analog-digital simulation)
  of communication system, **4**-12
  digital, **2**-1
  of digital computer, **2**-2
  of nonlinearities, **8**-18 to **8**-20
  space-vehicle, **2**-5, **8**-3
Simulation language, **9**-15
Sine-wave oscillator, **4**-13, **6**-23, **8**-9*b*
Single-pole-double-throw switch, **5**-1*d* to **5**-4*a*
Six-diode bridge switch, **5**-2
Skewing error, **9**-2
Skip bus, **7**-10
Slack variable, **8**-10
Slew rate (*see* Rate limit)
Slope in function generator, **4**-15
Smoothing oscillator, **4**-15, **6**-3
Software for hybrid computer, **9**-11 to **9**-17
Space-vehicle simulation, **2**-5, **8**-3
Special-purpose computer, **7**-1
Square-root generation, **8**-9*b*
  (*See also* Inverse function)
Square-wave generator, **4**-12
Squaring network, **6**-2
Stability of operational amplifier, **3**-34
Static accuracy, **3**-2
Static check, **7**-6*c*, **9**-11
Static friction, **8**-19
Steepest descent, **8**-7*b*
  division, **8**-9*b*
  square-root generation, **8**-9*b*
Step response, **3**-2
Step-size determination, **8**-14*e*, **8**-15
Strip-chart recorder, **7**-5*b*
Subdevice bit, **7**-8
Summing amplifier, **1**-2, **1**-3
  errors, **3**-19
Summing integrator, **1**-2, **3**-20
Summing junction, **3**-10
Super-beta transistor, **3**-33
Switch:
  active, **5**-17
  nonsaturating, **5**-17
  (*See also* Current switching)
Switched amplifier, **5**-17
Switched-capacitance decoder, **5**-7
Switching curve, **8**-16*b*
Switching spike, **2**-4, **5**-1*e*, **9**-5*b*
  in digital-to-analog converters, **5**-9
System identification (*see* Model matching)

Tabletop computer, **7**-2
Temperature compensation:
  of amplifier drift, **3**-33
  of breakpoint drift, **4**-7
  in digital-to-analog converter, **5**-8
Termination of iteration, **8**-14*e*
Tests:
  multiplier, **6**-15
  operational amplifier, **3**-37, **3**-38
Time-delay generation, **9**-7, Appendix B
Time-optimal control, **8**-16*b*
Time-scale check, **7**-6*c*
Time-scale switch, **7**-3
Time scaling, **1**-6*b*, **2**-1, **2**-2
Time sharing:
  of analog computer, **7**-4*c*
  of hybrid computer, **9**-17
Timing circuits, **7**-3*b*
Timing errors, **3**-2*c*, **5**-16, **5**-17
  in hybrid computation, **9**-2
Track-hold circuit, **2**-2*c*, **5**-17, **8**-18
Track-hold pair, **2**-3

TRACK mode, **2**-2
Transcendental equation, **8**-8
Transdiode, **4**-16
Transfer characteristics, **4**-2
Transfer impedance, **1**-10, **5**-1*c*, **5**-6
  (*See also* Table in Appendix A)
Transfer operators:
  differential-analyzer techniques, **1**-8
  linear, **1**-7
  operational-amplifier methods, **1**-10
  for sampled data, **2**-4
  (*See also* Table in Appendix A)
Transformation of coordinates, **6**-17
Transistor limiter, **4**-8, **6**-20
Transistor switch, **5**-3, **5**-4*d*
  active, **5**-18
Translator, simulation-language, **9**-15
Transport delay, **9**-7, Appendix B
Trap, **7**-11
Triangular-waveform generator, **4**-12
Two's-complement code, **5**-6

University of Arizona, **3**-35, **5**-6, **5**-16, **6**-3, **7**-2*a*
Unloading circuit, **1**-10

Varistor function generator, **4**-17*b*
Vector criterion function, **8**-15
Vector resolution, **6**-17
Vector servo (*see* Inverse resolution)
Voltage regulator, **1**-10

Wien-bridge oscillator, **4**-3
Word counter, **7**-12*b*

Zero-order hold, **9**-3

Date Due

~~FEB 20 1974~~

~~JAN 1975~~

MAR 19 1976

~~NOV 15 1976~~

~~Dec 10/77~~

MAR 28 1981

APR

BRO DART CAT. NO. 23 233 PRINTED IN U.S.A.

TK 7888 .K67 1972
Korn, Granino Arthur, 192
Electronic analog and hybrid c
010101 000
0 1163 0171761 1
TRENT UNIVERSITY

TK7888 .K67 1972
Korn, Granino Arthur
Electronic analog and hubrid
computers

DATE	ISSUED TO

190913

190913

07-035363-8